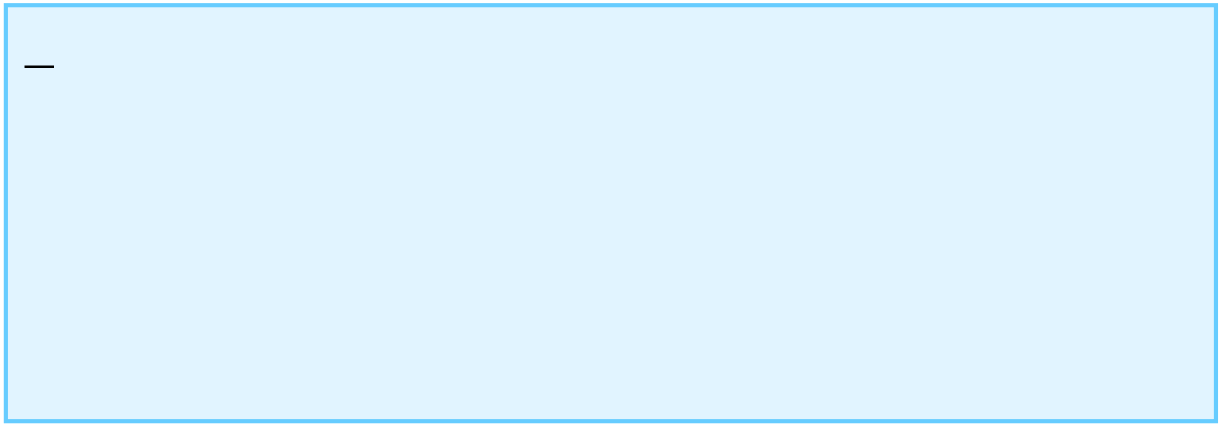
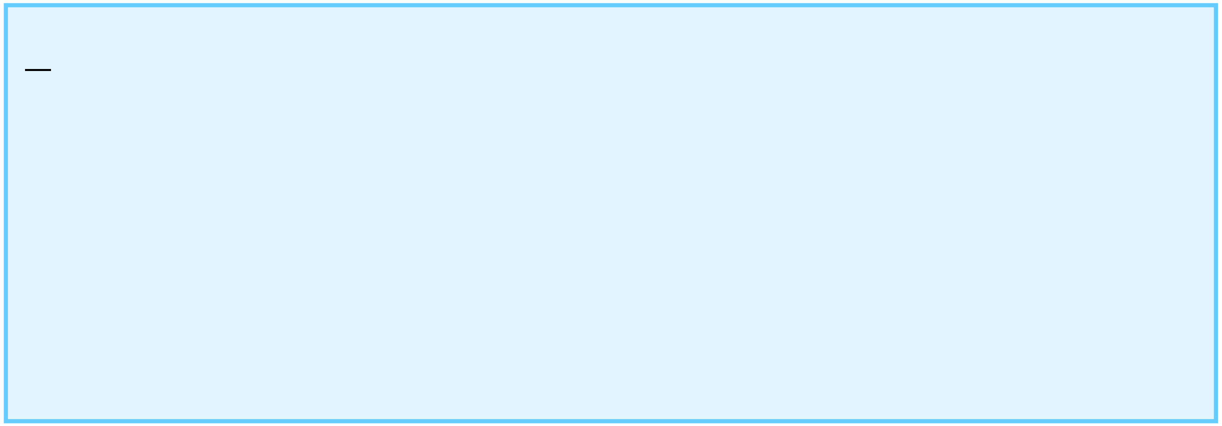
Ch1. Introduction to C++ Programming

```
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```



```
_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```



```
_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
W_
```

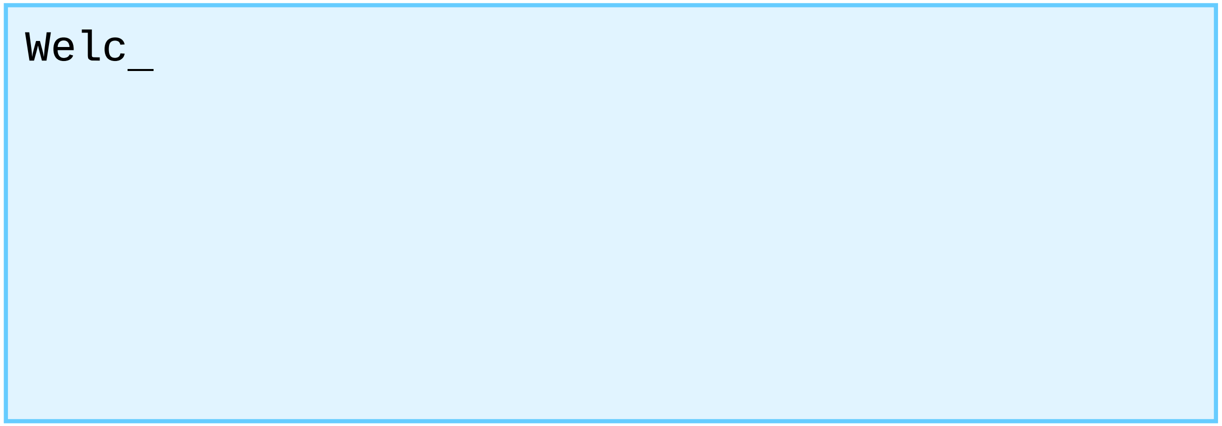
```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
We_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
Wel_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```



```
Welc_
```

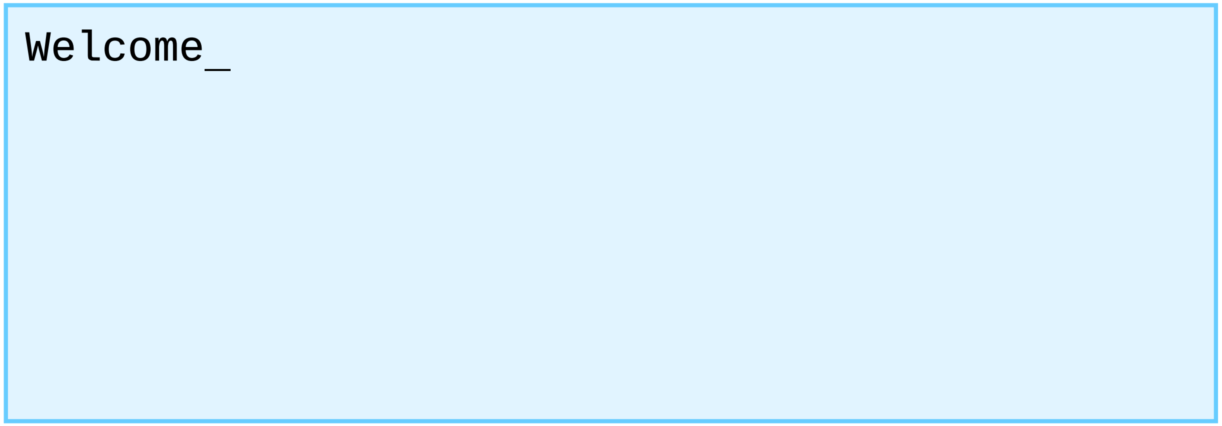
```
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
Welco_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

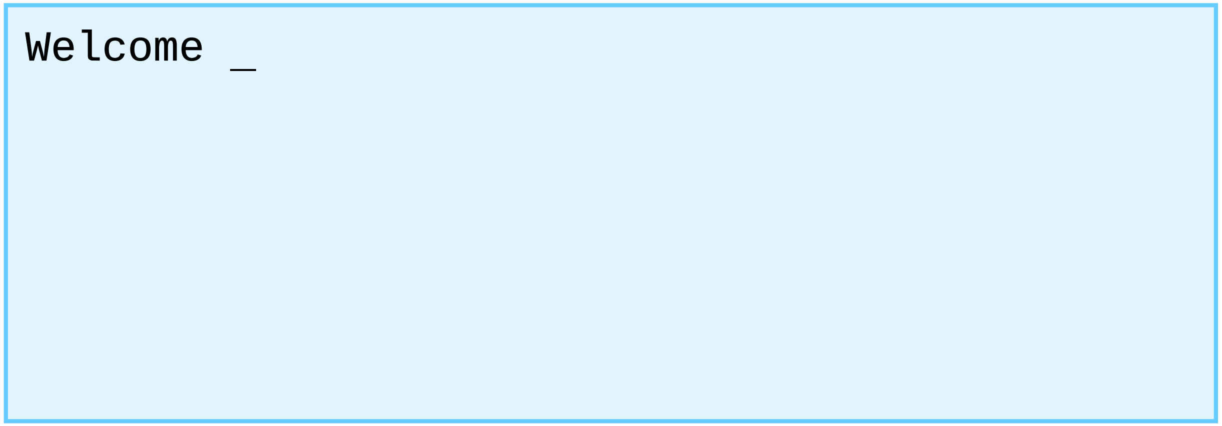
```
Welcom_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```



```
Welcome_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

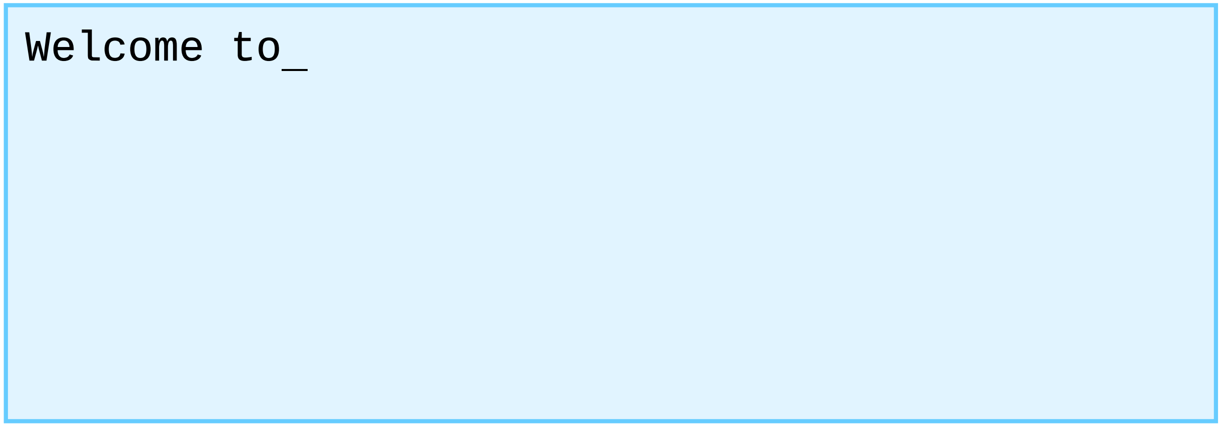


```
Welcome _
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
Welcome t_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```



```
Welcome to_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
Welcome to _
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
Welcome to Y_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
Welcome to Yu_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
Welcome to Yua_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
Welcome to Yuan_
```

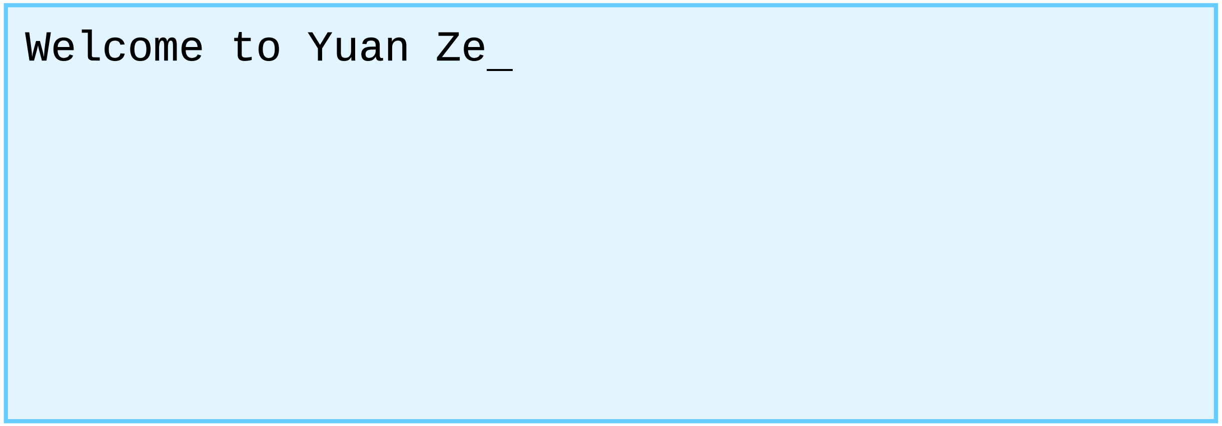
```
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
Welcome to Yuan _
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
Welcome to Yuan Z_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```



```
Welcome to Yuan Ze_
```

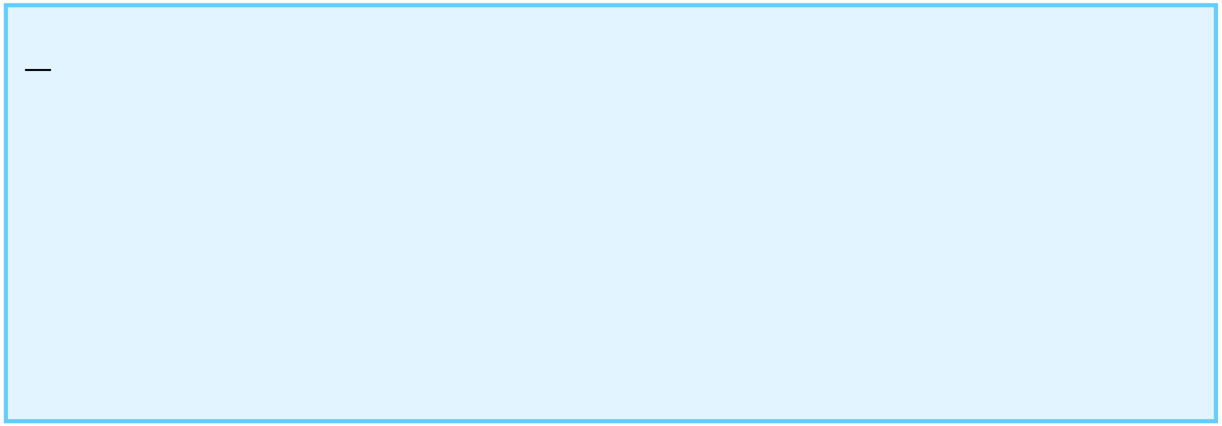
```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
Welcome to Yuan Ze!_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to Yuan Ze!\n";

   return 0;
}
```

```
Welcome to Yuan Ze!
_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```



```
_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

```
W_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

```
We_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

```
Wel_
```

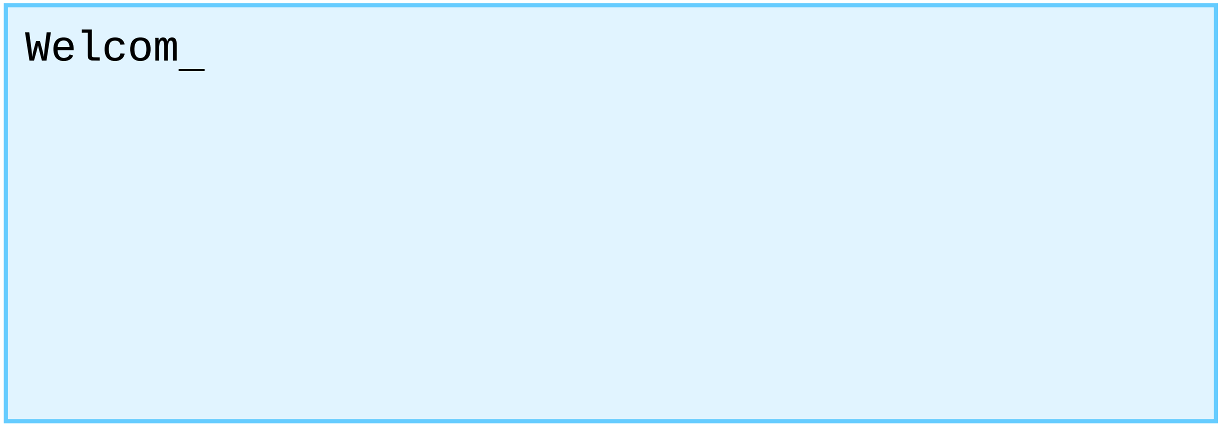
```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

```
Welc_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

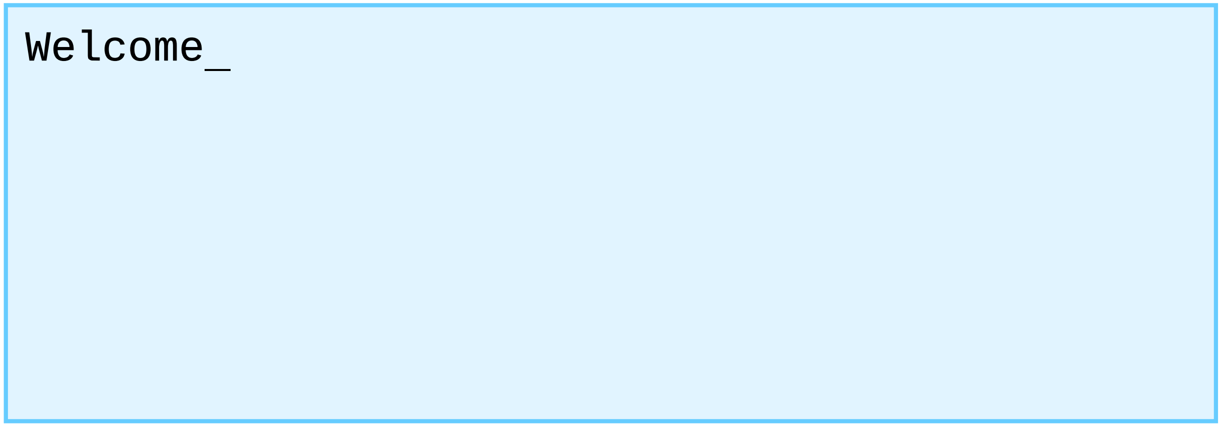
```
Welco_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```



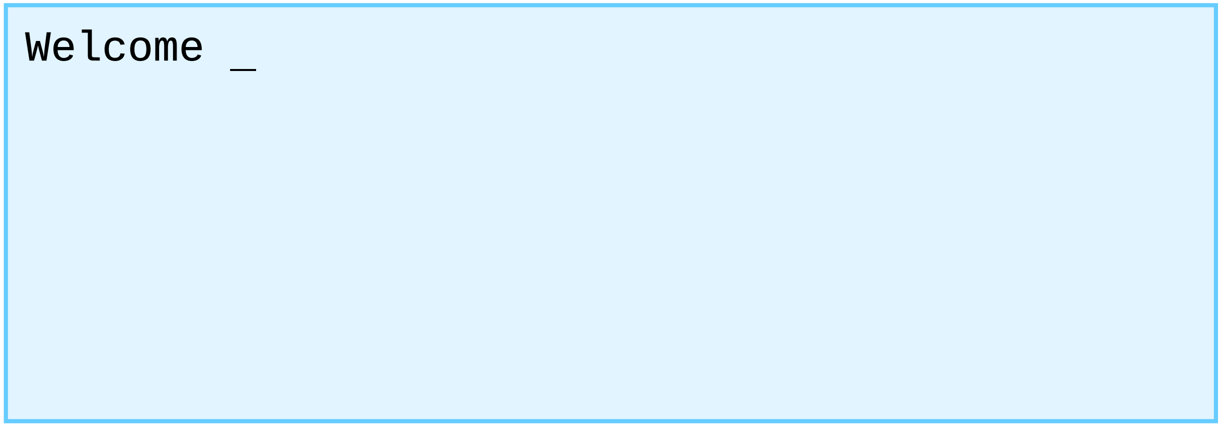
```
Welcom_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```



```
Welcome_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```



```
Welcome _
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

```
Welcome t_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

```
Welcome to_
```

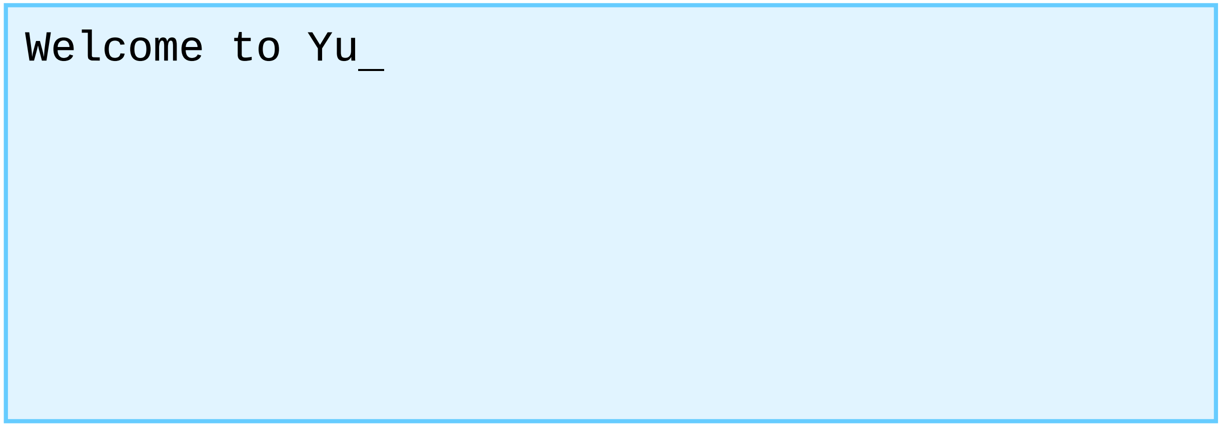
```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

```
Welcome to _
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

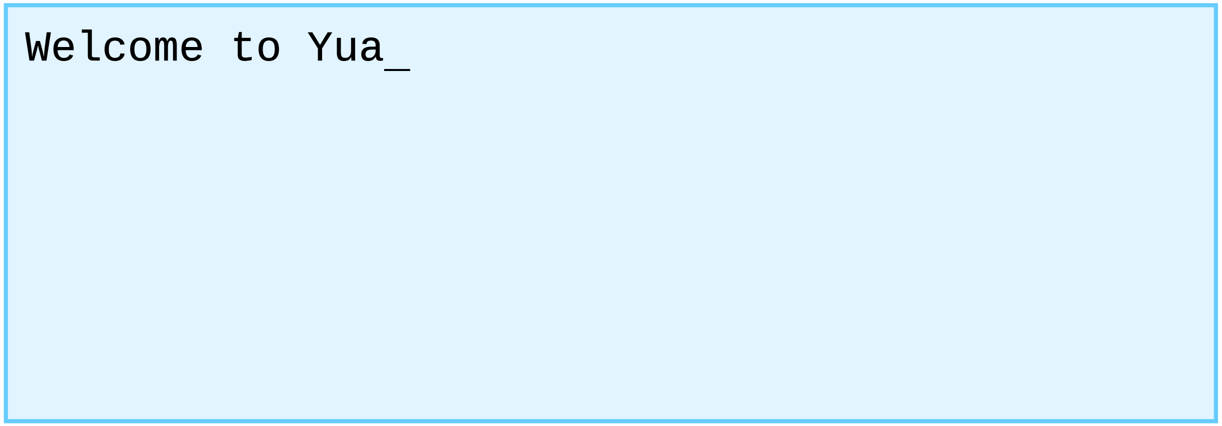
```
Welcome to Y_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```



```
Welcome to Yu_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

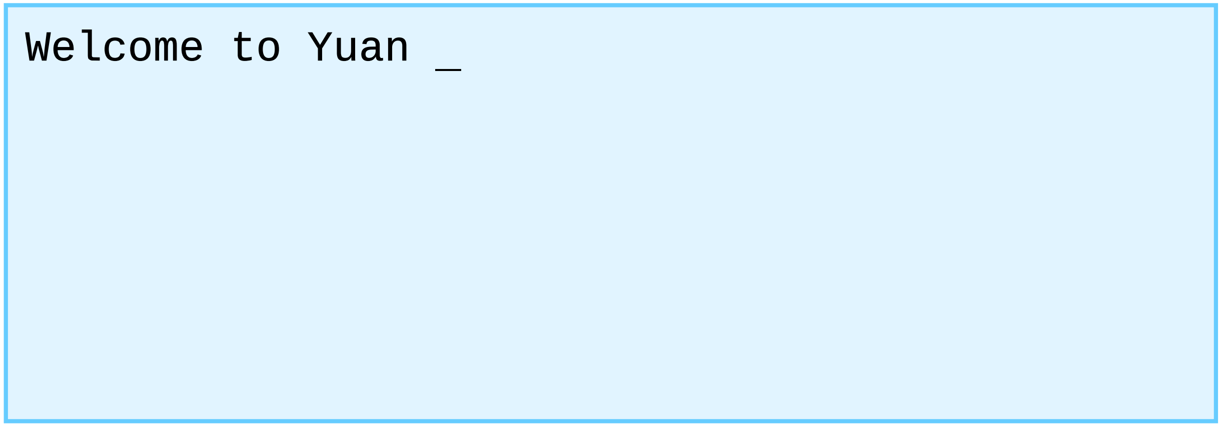


```
Welcome to Yua_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

```
Welcome to Yuan_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```



```
Welcome to Yuan _
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

```
Welcome to Yuan Z_
```

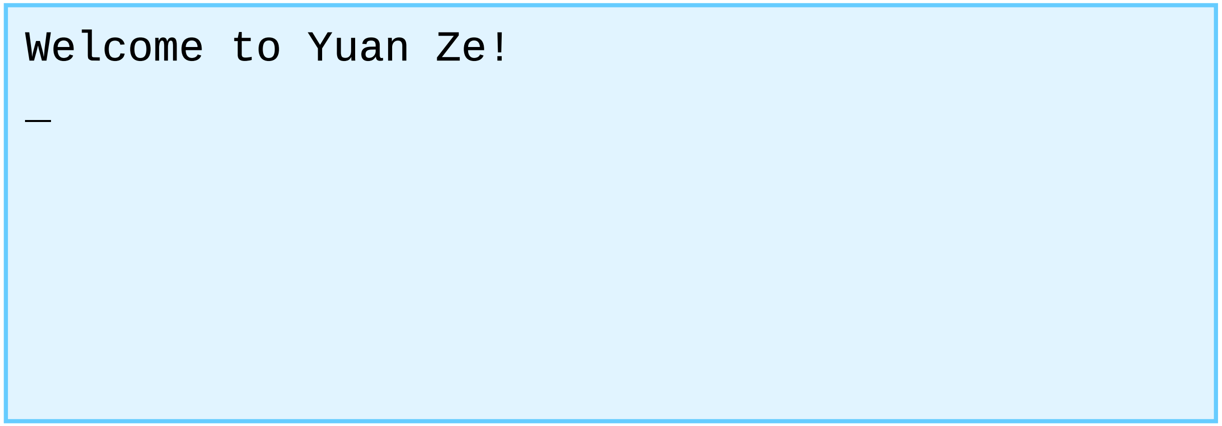
```
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

```
Welcome to Yuan Ze_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```

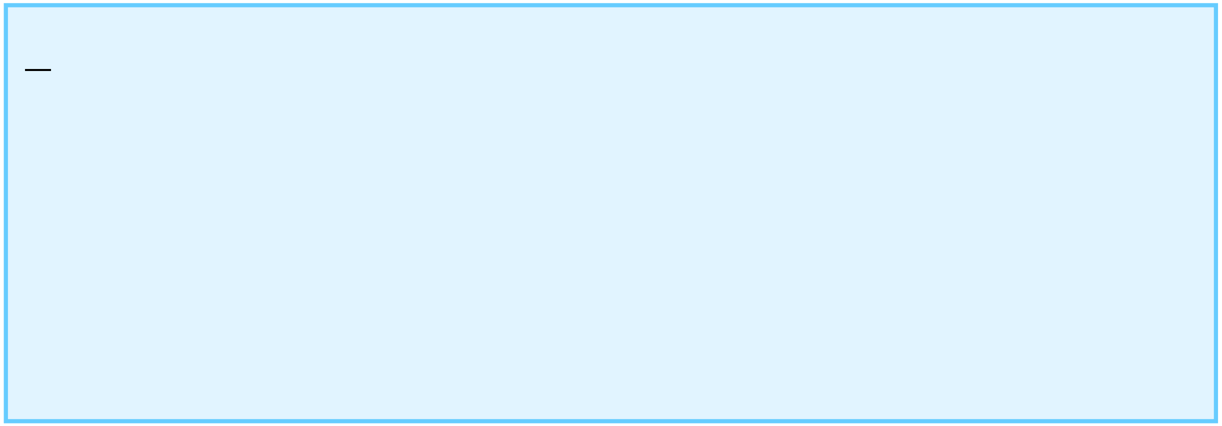
```
Welcome to Yuan Ze!_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to ";
   std::cout << "Yuan Ze!\n";

   return 0;
}
```



```
Welcome to Yuan Ze!
_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```



```
_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
W_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
We_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
Wel_
```

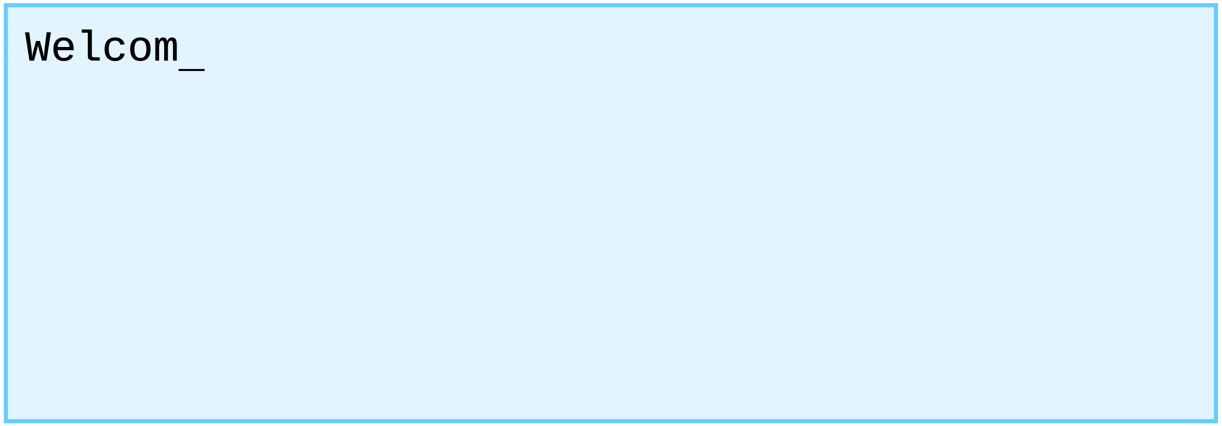
```
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
Welc_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
Welco_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```



```
Welcom_
```

```cpp
#include <iostream>

int main()
{
    std::cout << "Welcome to\nYuan\n\nZe!\n";

    return 0;
}
```

```
Welcome_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
Welcome _
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
Welcome t_
```

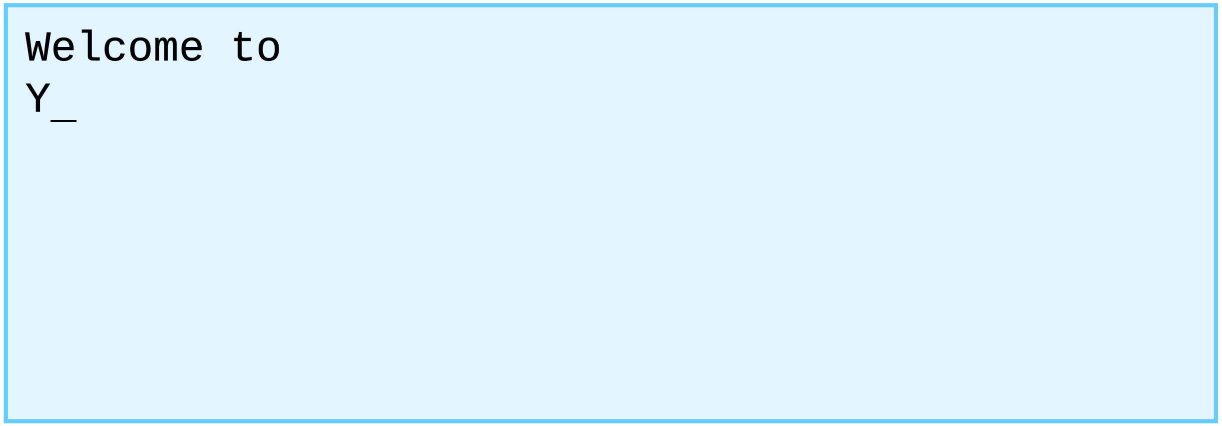
```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
Welcome to_
```

```cpp
#include <iostream>

int main()
{
    std::cout << "Welcome to\nYuan\n\nZe!\n";

    return 0;
}
```

```
Welcome to
_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```



```
Welcome to
Y_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
Welcome to
Yu_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
Welcome to
Yua_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
Welcome to
Yuan_
```

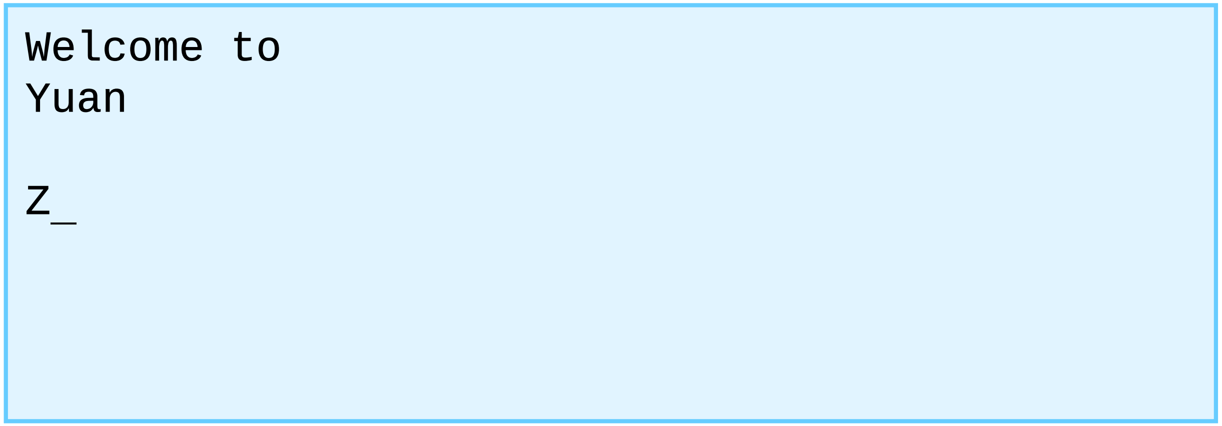
```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
Welcome to
Yuan
_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

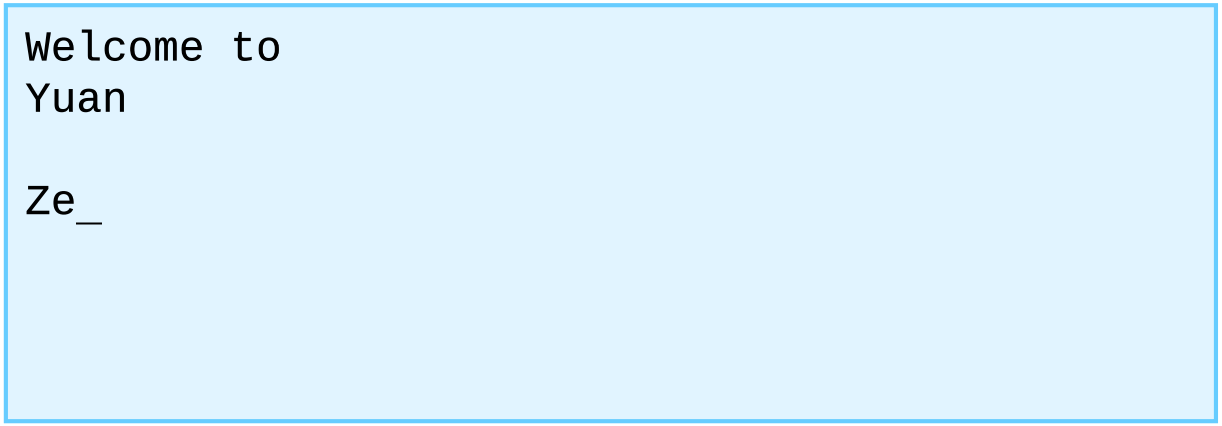
```
Welcome to
Yuan

_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```



```
Welcome to
Yuan

Z_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```



```
Welcome to
Yuan

Ze_
```

```
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
Welcome to
Yuan

Ze!_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Welcome to\nYuan\n\nZe!\n";

   return 0;
}
```

```
Welcome to
Yuan

Ze!

_
```

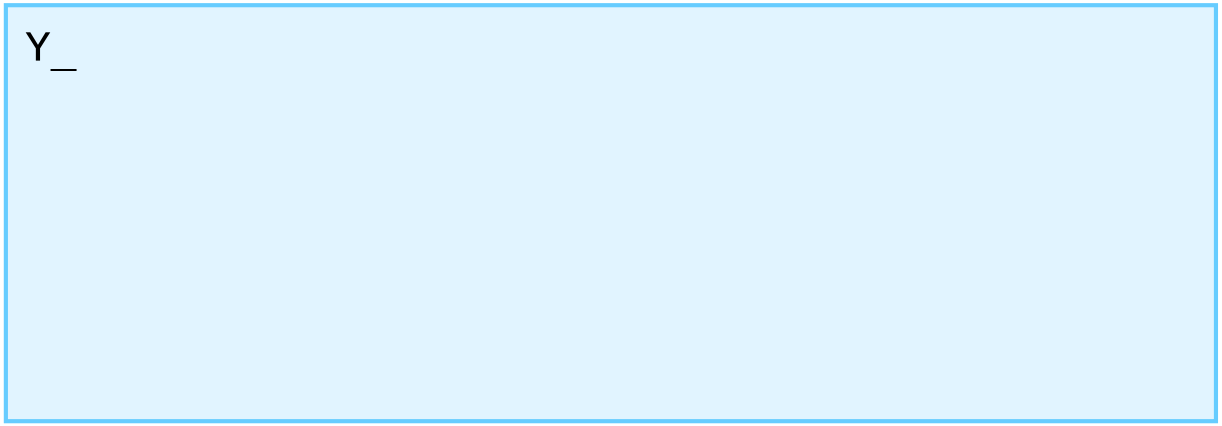
Escape sequences

Escape sequence	Description
`\'`	single quote
`\"`	double quote
`\?`	question mark
`\\`	backslash
`\a`	audible bell
`\b`	backspace
`\n`	line feed - new line
`\r`	carriage return
`\t`	horizontal tab

```cpp
#include <iostream>

int main()
{
   std::cout << "Yuan\bZe\n";

   return 0;
}
```

```
_
```

```cpp
#include <iostream>

int main()
{
    std::cout << "Yuan\bZe\n";

    return 0;
}
```



```
Y_
```

```cpp
#include <iostream>

int main()
{
    std::cout << "Yuan\bZe\n";

    return 0;
}
```

```
Yu_
```

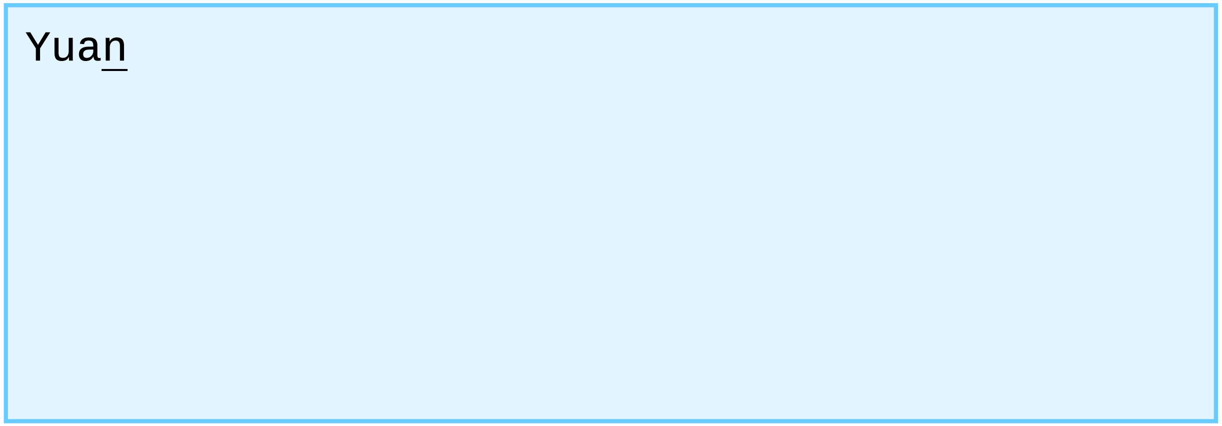
```cpp
#include <iostream>

int main()
{
   std::cout << "Yuan\bZe\n";

   return 0;
}
```

```
Yua_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Yuan\bZe\n";

   return 0;
}
```

```
Yuan_
```

```cpp
#include <iostream>

int main()
{
    std::cout << "Yuan\bZe\n";

    return 0;
}
```

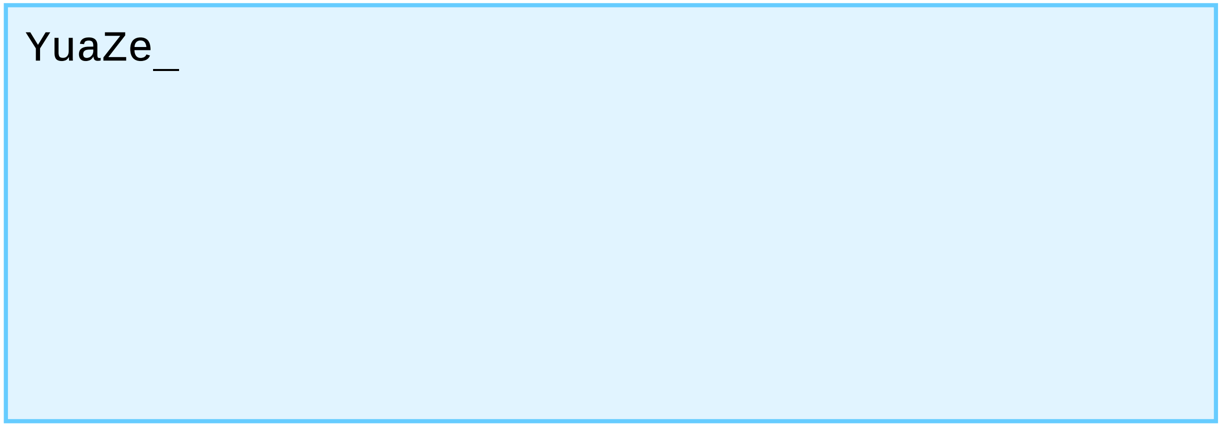


```
Yuan
```

```
#include <iostream>

int main()
{
   std::cout << "Yuan\bZe\n";

   return 0;
}
```

```
YuaZ_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Yuan\bZe\n";

   return 0;
}
```



```
YuaZe_
```

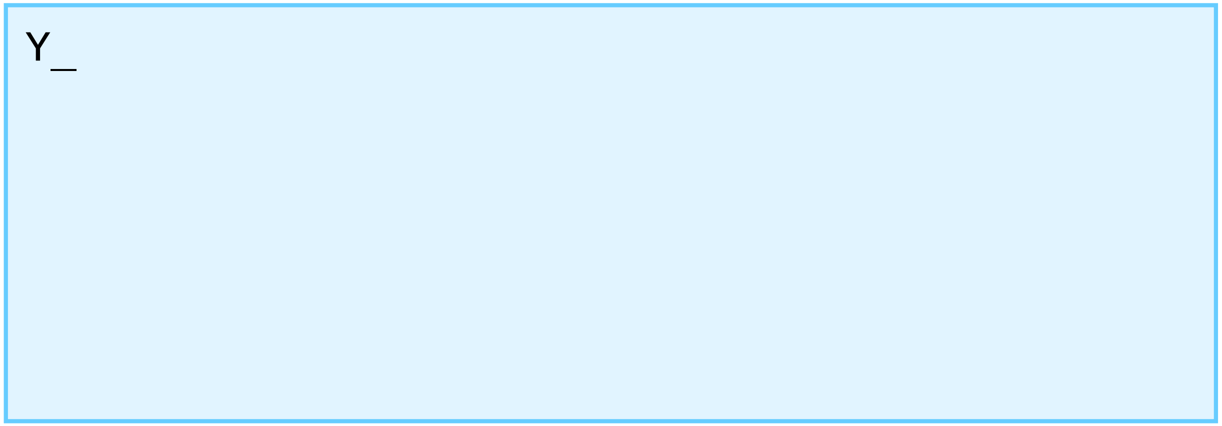
```cpp
#include <iostream>

int main()
{
   std::cout << "Yuan\bZe\n";

   return 0;
}
```

```
YuaZe
_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Yuan\rZe\n";

   return 0;
}
```

```
_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Yuan\rZe\n";

   return 0;
}
```



```
Y_
```

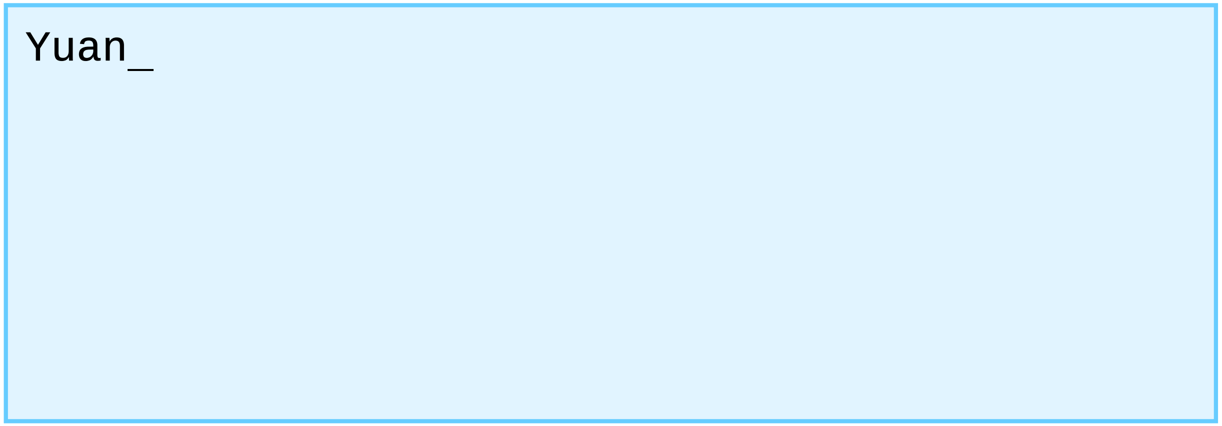
```cpp
#include <iostream>

int main()
{
   std::cout << "Yuan\rZe\n";

   return 0;
}
```

```
Yu_
```

```
#include <iostream>

int main()
{
   std::cout << "Yuan\rZe\n";

   return 0;
}
```

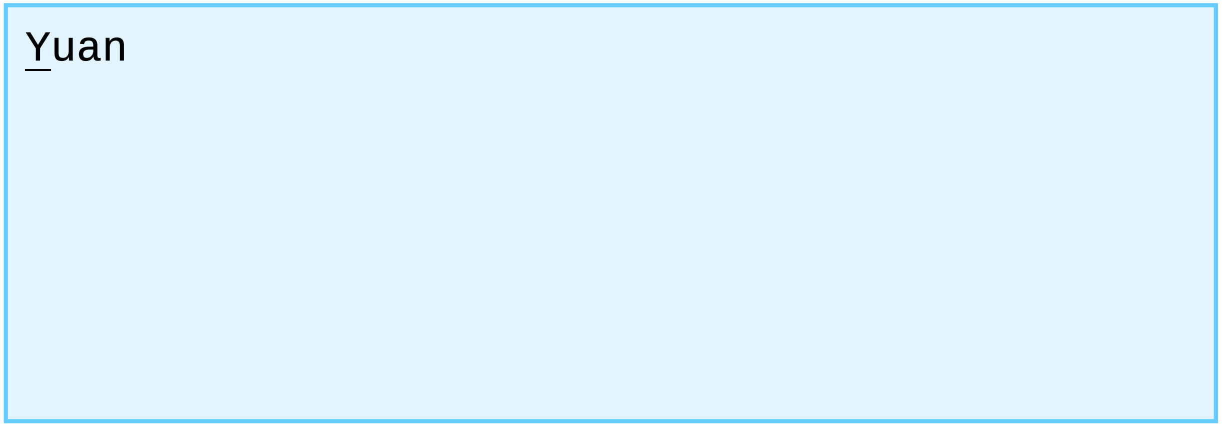
```
Yua_
```

```cpp
#include <iostream>

int main()
{
   std::cout << "Yuan\rZe\n";

   return 0;
}
```



```
Yuan_
```

```cpp
#include <iostream>

int main()
{
    std::cout << "Yuan\rZe\n";

    return 0;
}
```



```
Yuan
```

```cpp
#include <iostream>

int main()
{
    std::cout << "Yuan\rZe\n";

    return 0;
}
```

```
Zuan
```

```
#include <iostream>

int main()
{
   std::cout << "Yuan\rZe\n";

   return 0;
}
```

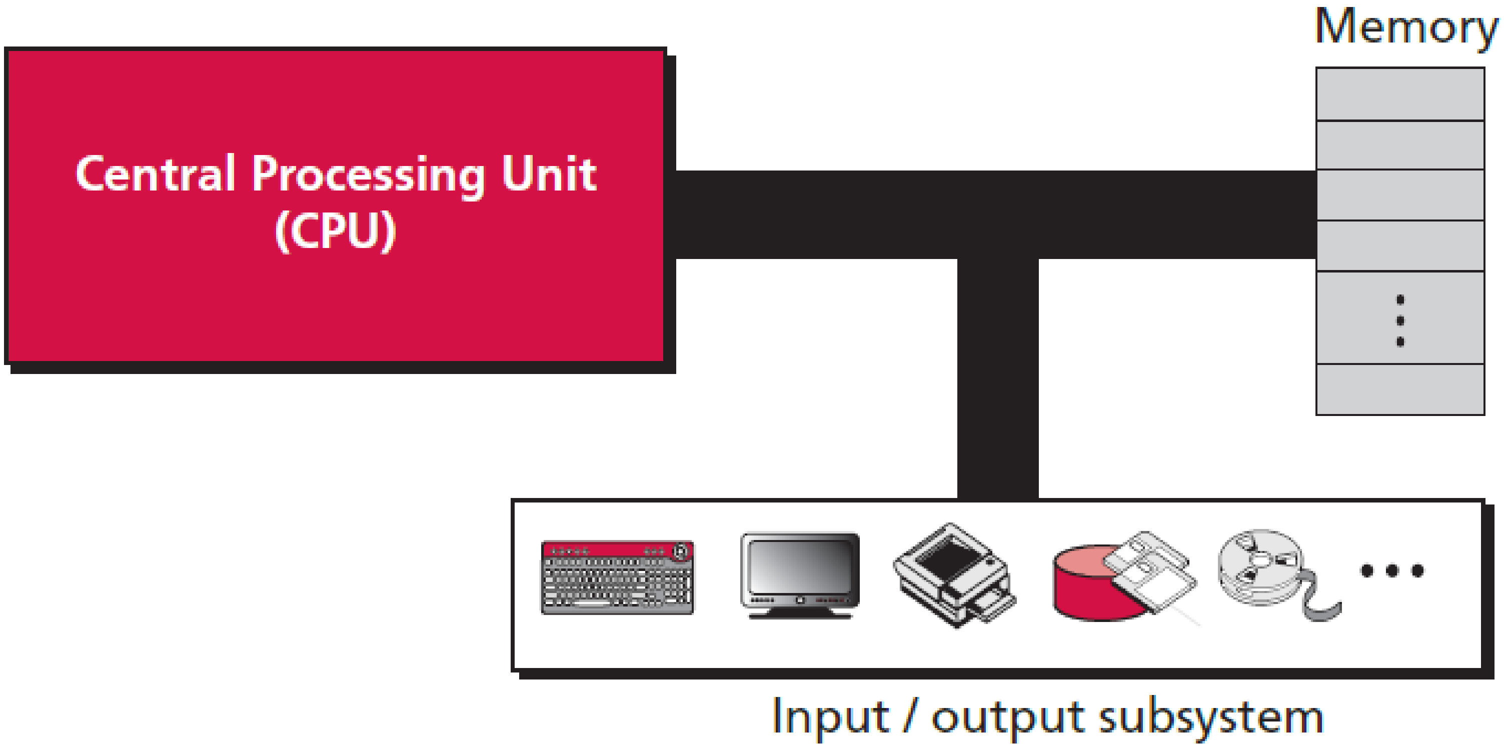
```
Zean
```

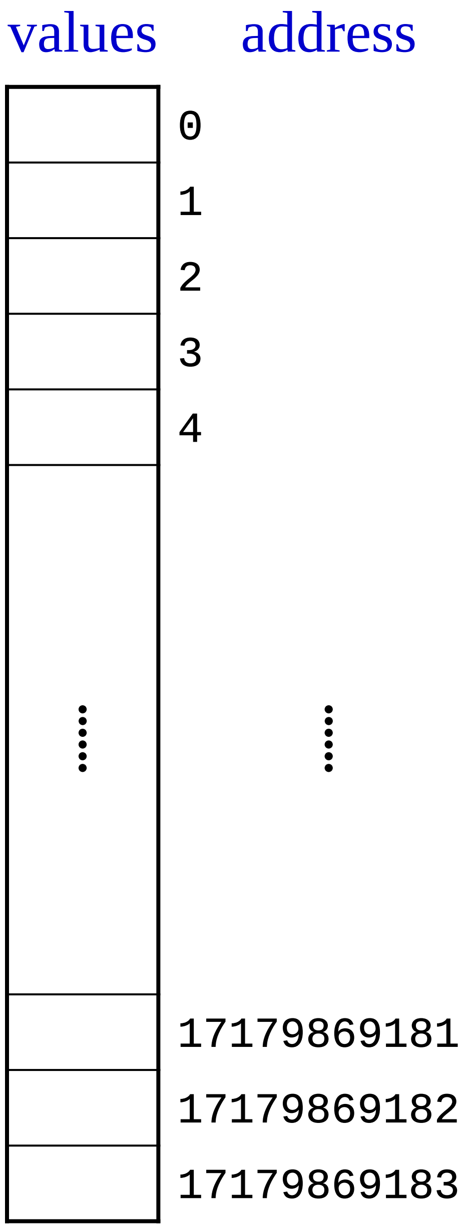
```
#include <iostream>

int main()
{
   std::cout << "Yuan\rZe\n";

   return 0;
}
```

```
Zean
_
```

```cpp
#include <iostream>

int main()
{
   int integer1;
   int integer2;
   int sum;

   std::cout << "Enter first integer: ";
   std::cin >> integer1;

   std::cout << "Enter second integer: ";
   std::cin >> integer2;

   sum = integer1 + integer2;

   std::cout << "Sum is " << sum << std::endl;
}
```

Computer Hardware

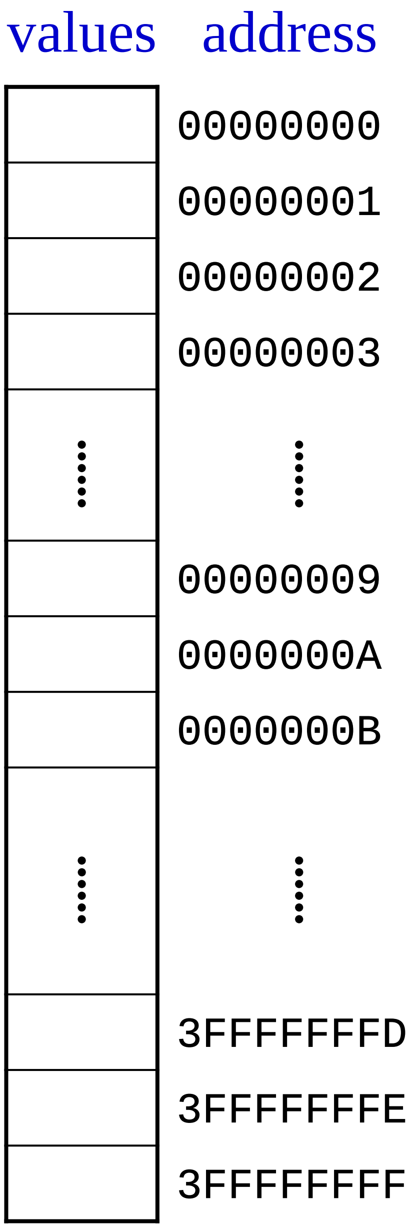




values address

0
1
2
3
4
⋮ ⋮
17179869181
17179869182
17179869183

Main Memory (16GB)



values address

00000000
00000001
00000002
00000003
⋮ ⋮
00000009
0000000A
0000000B
⋮ ⋮
3FFFFFFD
3FFFFFFE
3FFFFFFF

binary representation	hexadecimal	decimal
00000000 00000000 00000000 00000000	00000000	0
00000000 00000000 00000000 00000001	00000001	1
00000000 00000000 00000000 00000010	00000002	2
00000000 00000000 00000000 00000011	00000003	3
00000000 00000000 00000000 00000100	00000004	4
00000000 00000000 00000000 00000101	00000005	5
00000000 00000000 00000000 00000110	00000006	6
00000000 00000000 00000000 00000111	00000007	7
00000000 00000000 00000000 00001000	00000008	8
00000000 00000000 00000000 00001001	00000009	9
00000000 00000000 00000000 00001010	0000000A	10
00000000 00000000 00000000 00001011	0000000B	11
00000000 00000000 00000000 00001100	0000000C	12
00000000 00000000 00000000 00001101	0000000D	13
00000000 00000000 00000000 00001110	0000000E	14
00000000 00000000 00000000 00001111	0000000F	15

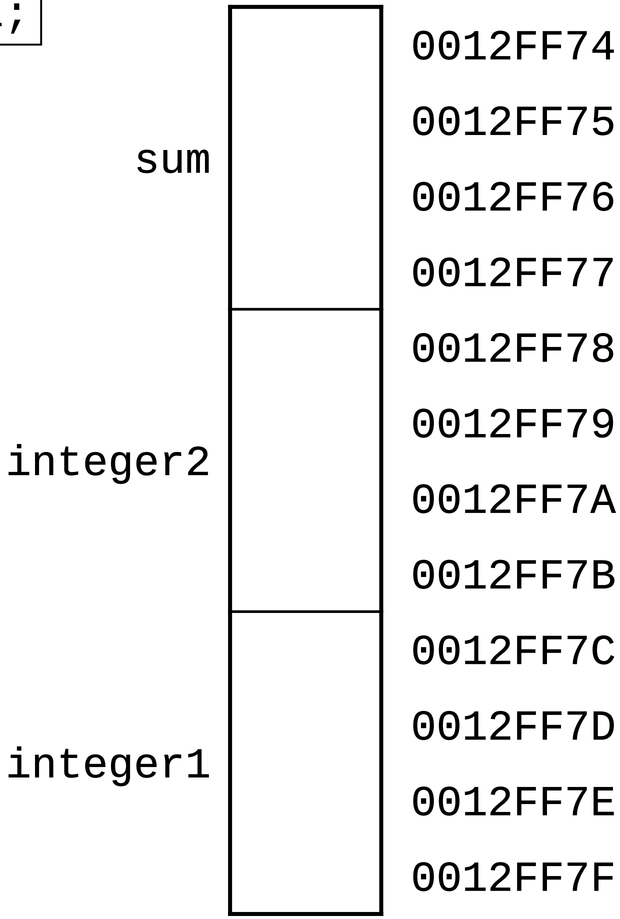
binary representation	hexadecimal	decimal
00000000 00000000 00000000 00010000	00000010	16
00000000 00000000 00000000 00010001	00000011	17
00000000 00000000 00000000 00010010	00000012	18
00000000 00000000 00000000 00010011	00000013	19
00000000 00000000 00000000 00010100	00000014	20
00000000 00000000 00000000 00010101	00000015	21
00000000 00000000 00000000 00010110	00000016	22
00000000 00000000 00000000 00010111	00000017	23
00000000 00000000 00000000 00011000	00000018	24
00000000 00000000 00000000 00011001	00000019	25
00000000 00000000 00000000 00011010	0000001A	26
00000000 00000000 00000000 00011011	0000001B	27
00000000 00000000 00000000 00011100	0000001C	28
00000000 00000000 00000000 00011101	0000001D	29
00000000 00000000 00000000 00011110	0000001E	30
00000000 00000000 00000000 00011111	0000001F	31

binary representation	hexadecimal	decimal
00000000 00000000 00000000 00100000	00000020	32
00000000 00000000 00000000 00100001	00000021	33
00000000 00000000 00000000 00100010	00000022	34
00000000 00000000 00000000 00100011	00000023	35
. . . .	. . . .	. . . .
00000000 00010010 11111111 01110100	0012FF74	1245044
00000000 00010010 11111111 01111000	0012FF78	1245048
00000000 00010010 11111111 01111100	0012FF7C	1245052
00000000 00010010 11111111 10000000	0012FF80	1245056
. . . .	. . . .	. . . .
11111111 11111111 11111111 11111010	FFFFFFFA	4294967290
11111111 11111111 11111111 11111011	FFFFFFFB	4294967291
11111111 11111111 11111111 11111100	FFFFFFFC	4294967292
11111111 11111111 11111111 11111101	FFFFFFFD	4294967293
11111111 11111111 11111111 11111110	FFFFFFFE	4294967294
11111111 11111111 11111111 11111111	FFFFFFFF	4294967295

```
std::cout << "Enter first integer: ";
std::cin >> integer1;
std::cout << "Enter second integer: ";
std::cin >> integer2;
sum = integer1 + integer2;
std::cout << "Sum is " << sum << std::endl;
```

Output

Memory

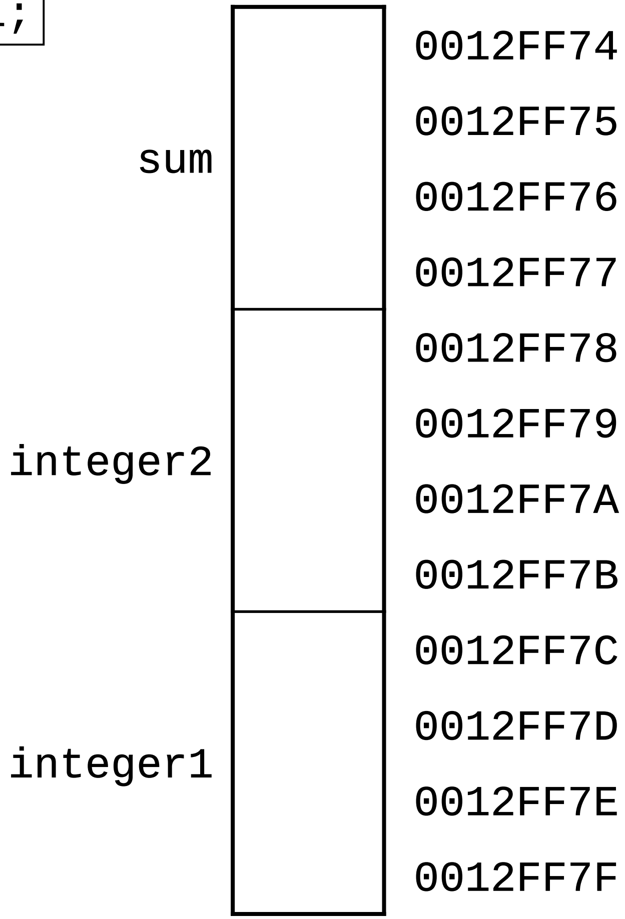


```
std::cout << "Enter first integer: ";
std::cin >> integer1;
std::cout << "Enter second integer: ";
std::cin >> integer2;
sum = integer1 + integer2;
std::cout << "Sum is " << sum << std::endl;
```

```
Enter first integer: _
```

Output

Memory

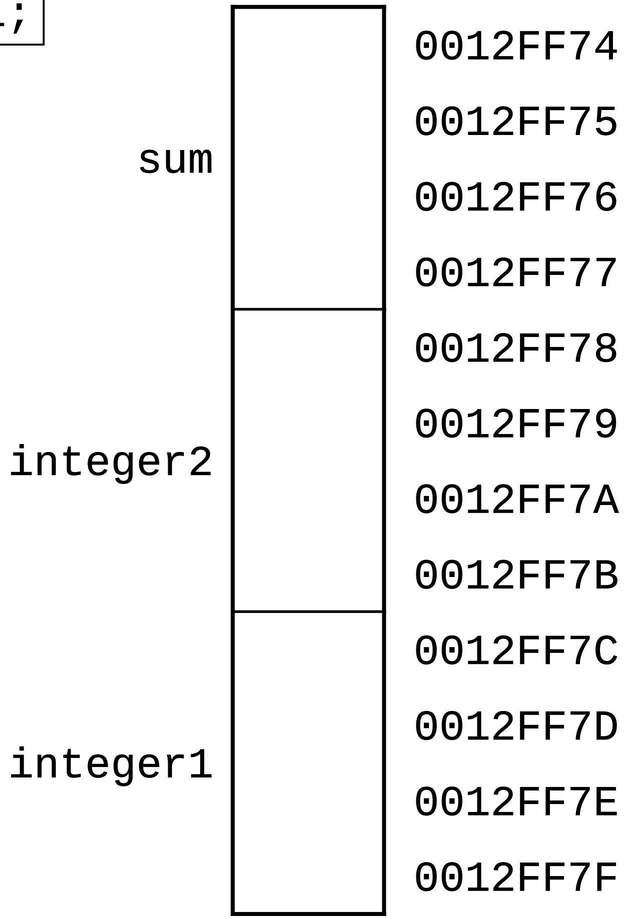


```
std::cout << "Enter first integer: ";
std::cin >> integer1;
std::cout << "Enter second integer: ";
std::cin >> integer2;
sum = integer1 + integer2;
std::cout << "Sum is " << sum << std::endl;
```

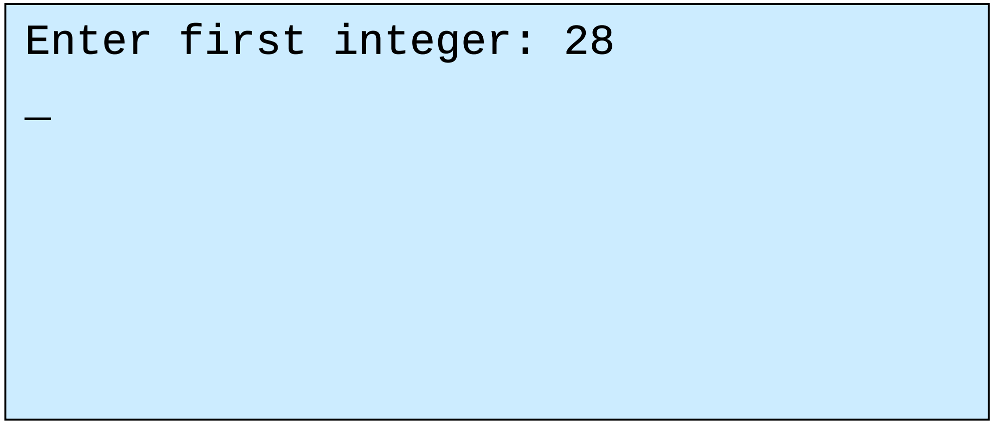
```
Enter first integer: 28_
```

Output

Memory



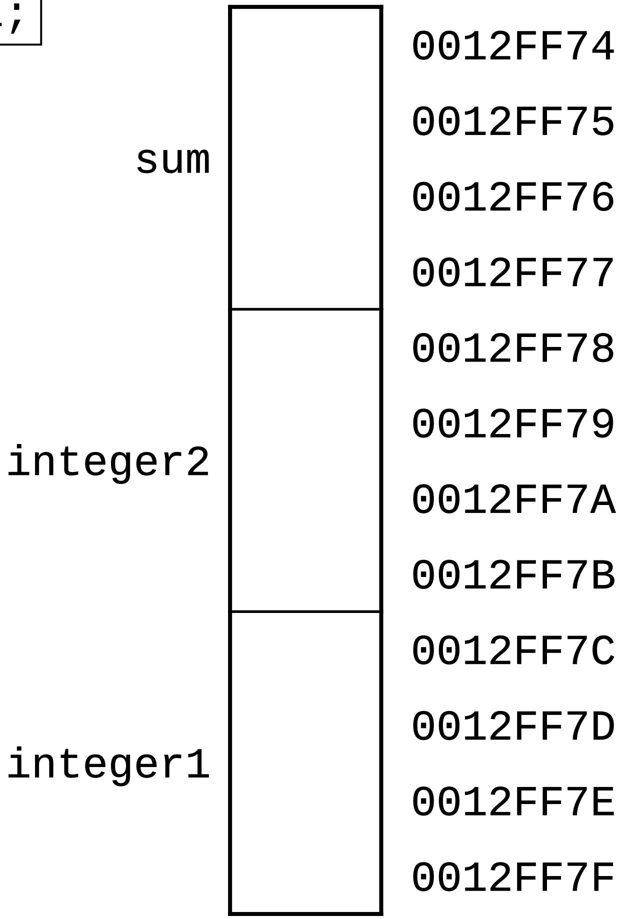
```
std::cout << "Enter first integer: ";
std::cin >> integer1;
std::cout << "Enter second integer: ";
std::cin >> integer2;
sum = integer1 + integer2;
std::cout << "Sum is " << sum << std::endl;
```



```
Enter first integer: 28
_
```

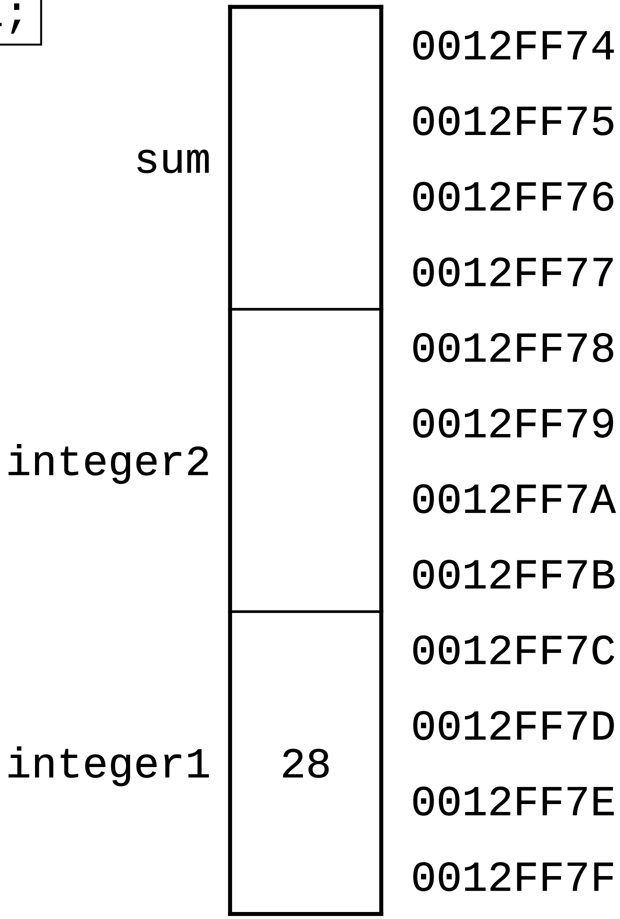
Output

Memory



```
std::cout << "Enter first integer: ";
std::cin >> integer1;
std::cout << "Enter second integer: ";
std::cin >> integer2;
sum = integer1 + integer2;
std::cout << "Sum is " << sum << std::endl;
```

Memory



```
Enter first integer: 28
_
```

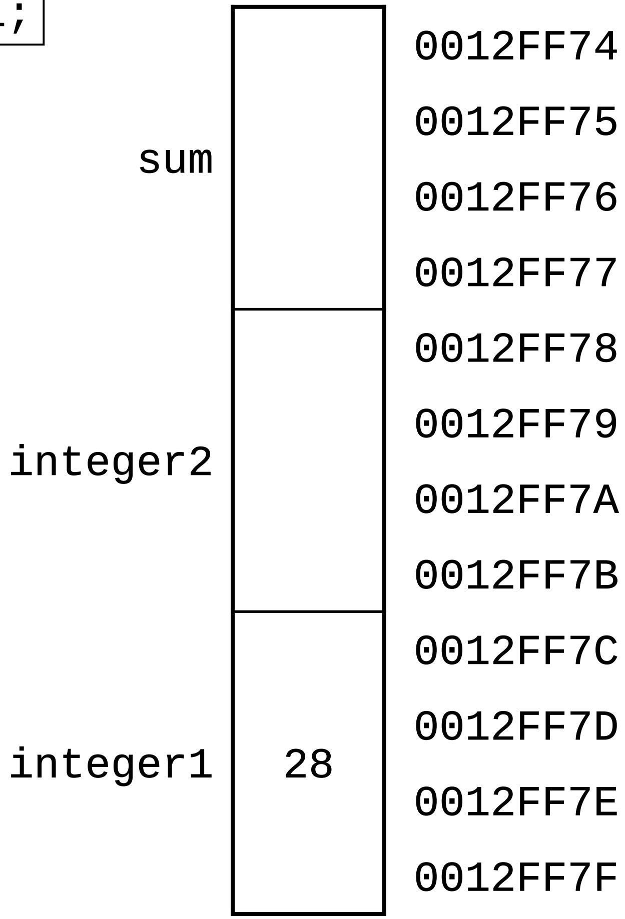
Output

```
std::cout << "Enter first integer: ";
std::cin >> integer1;
std::cout << "Enter second integer: ";
std::cin >> integer2;
sum = integer1 + integer2;
std::cout << "Sum is " << sum << std::endl;
```

```
Enter first integer: 28
Enter second integer: _
```

Output

Memory



sum

integer2

integer1

28

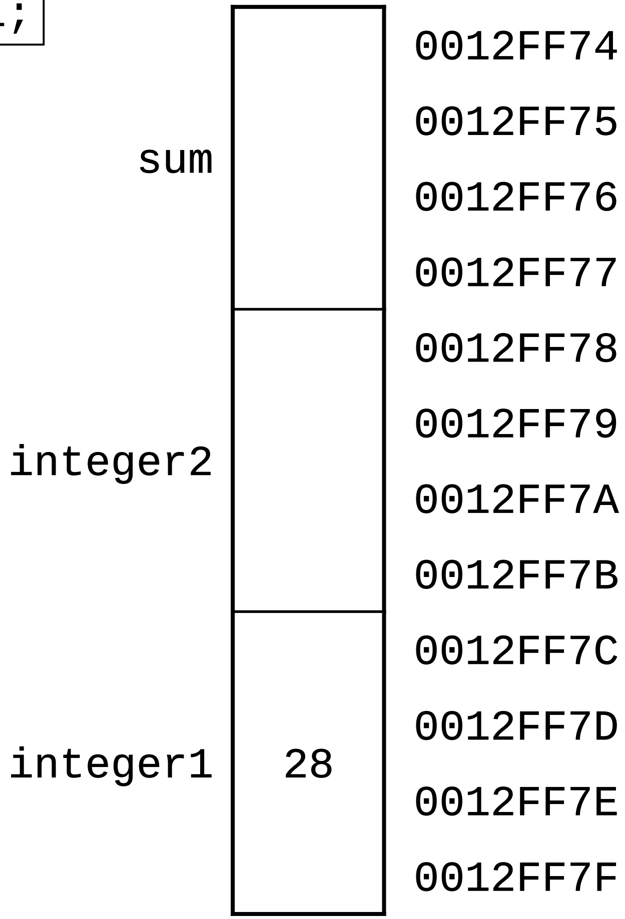
0012FF74
0012FF75
0012FF76
0012FF77
0012FF78
0012FF79
0012FF7A
0012FF7B
0012FF7C
0012FF7D
0012FF7E
0012FF7F

```cpp
std::cout << "Enter first integer: ";
std::cin >> integer1;
std::cout << "Enter second integer: ";
std::cin >> integer2;
sum = integer1 + integer2;
std::cout << "Sum is " << sum << std::endl;
```

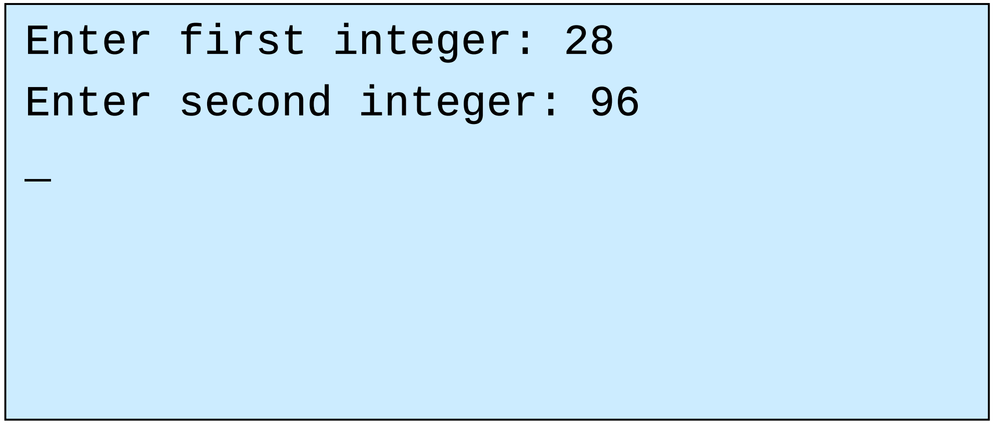
```
Enter first integer: 28
Enter second integer: 96_
```

Output

Memory



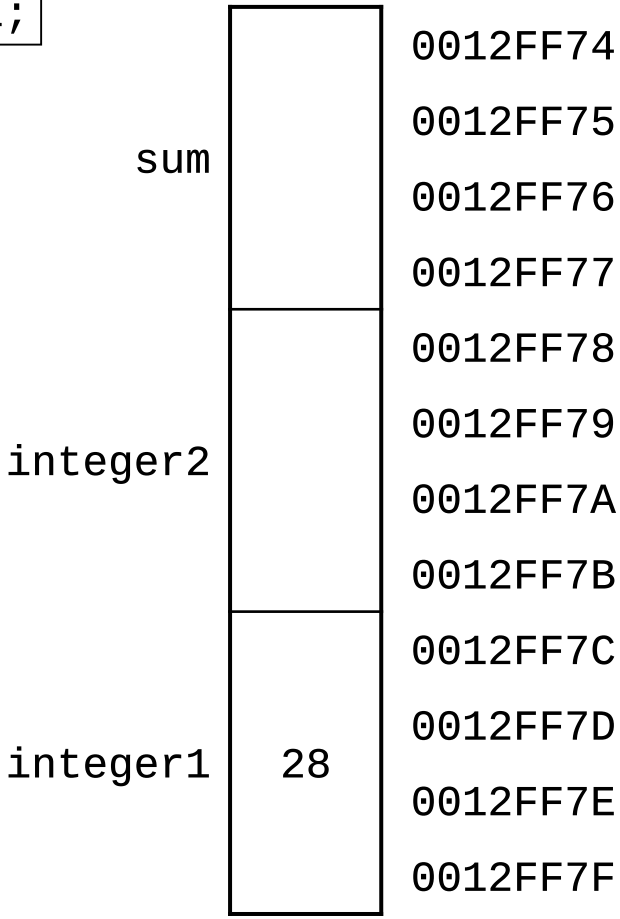
```
std::cout << "Enter first integer: ";
std::cin >> integer1;
std::cout << "Enter second integer: ";
std::cin >> integer2;
sum = integer1 + integer2;
std::cout << "Sum is " << sum << std::endl;
```



```
Enter first integer: 28
Enter second integer: 96
_
```

Output

Memory



sum

integer2

integer1

28

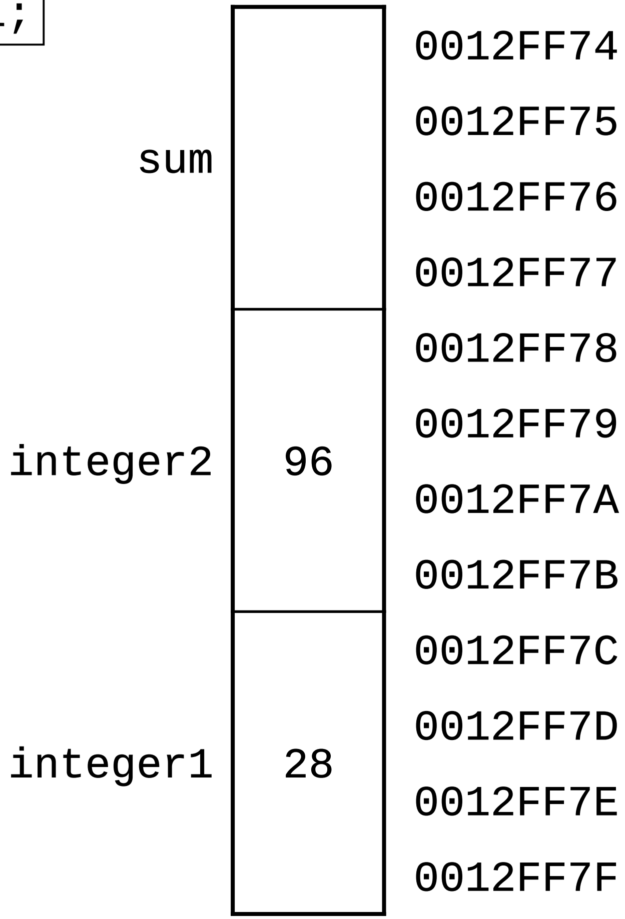
0012FF74
0012FF75
0012FF76
0012FF77
0012FF78
0012FF79
0012FF7A
0012FF7B
0012FF7C
0012FF7D
0012FF7E
0012FF7F

```
std::cout << "Enter first integer: ";
std::cin >> integer1;
std::cout << "Enter second integer: ";
std::cin >> integer2;
sum = integer1 + integer2;
std::cout << "Sum is " << sum << std::endl;
```

```
Enter first integer: 28
Enter second integer: 96
_
```

Output

Memory



sum

integer2 96

integer1 28

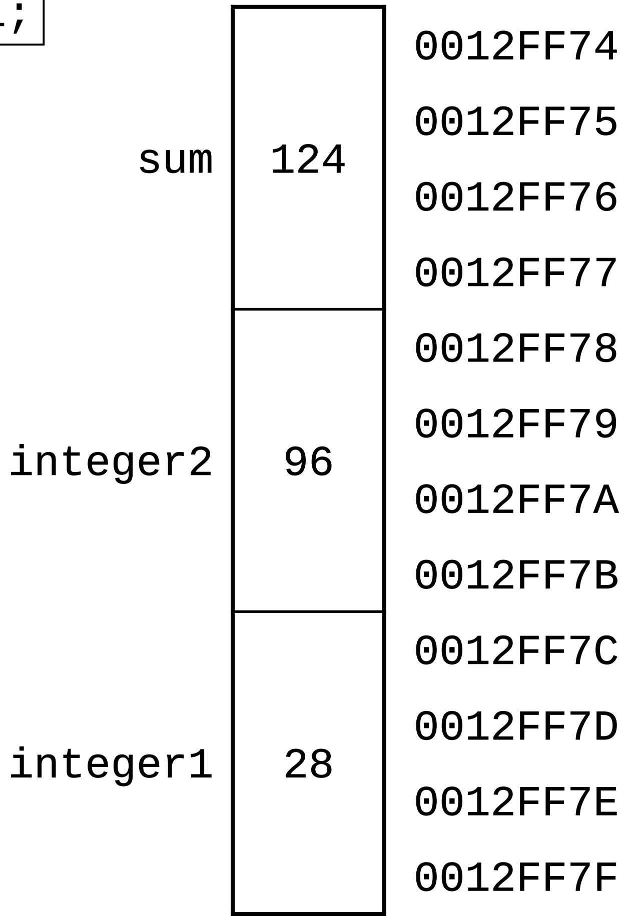
0012FF74
0012FF75
0012FF76
0012FF77
0012FF78
0012FF79
0012FF7A
0012FF7B
0012FF7C
0012FF7D
0012FF7E
0012FF7F

```
std::cout << "Enter first integer: ";
std::cin >> integer1;
std::cout << "Enter second integer: ";
std::cin >> integer2;
sum = integer1 + integer2;
std::cout << "Sum is " << sum << std::endl;
```

```
Enter first integer: 28
Enter second integer: 96
_
```

Output

Memory

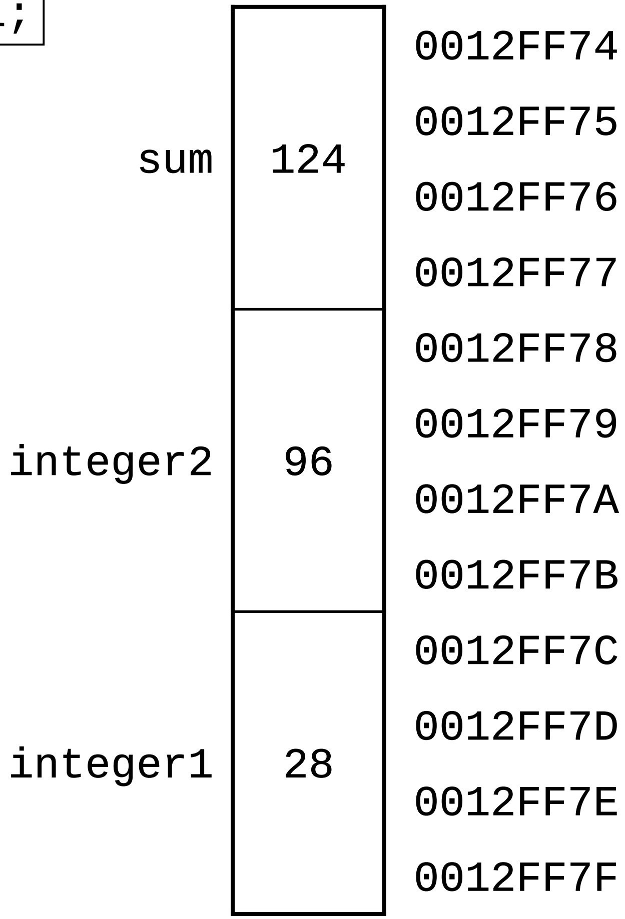


```
std::cout << "Enter first integer: ";
std::cin >> integer1;
std::cout << "Enter second integer: ";
std::cin >> integer2;
sum = integer1 + integer2;
std::cout << "Sum is " << sum << std::endl;
```

```
Enter first integer: 28
Enter second integer: 96
Sum is 124_
```

Output

Memory

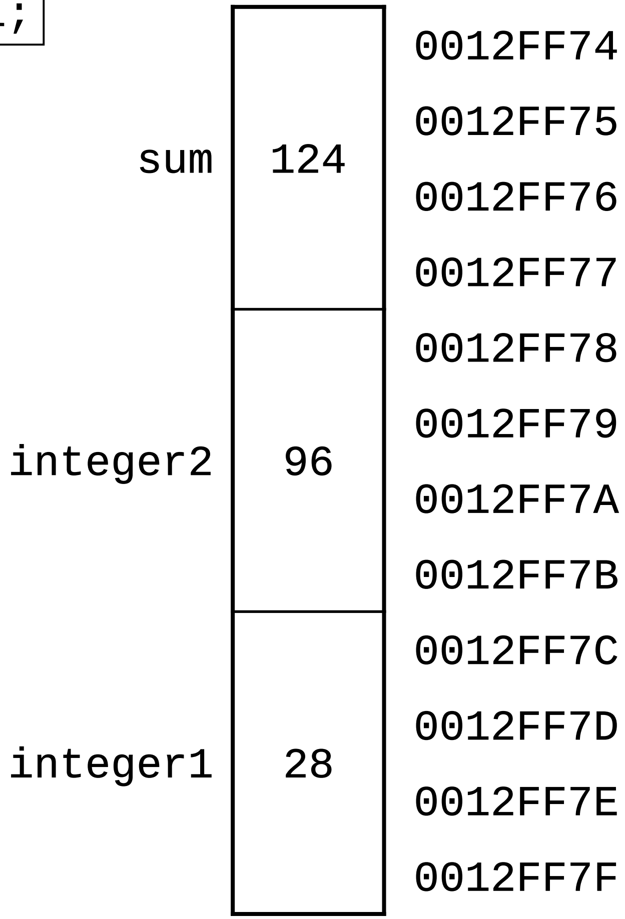


```cpp
std::cout << "Enter first integer: ";
std::cin >> integer1;
std::cout << "Enter second integer: ";
std::cin >> integer2;
sum = integer1 + integer2;
std::cout << "Sum is " << sum << std::endl;
```

```
Enter first integer: 28
Enter second integer: 96
Sum is 124
_
```

Output

Memory



sum 124

integer2 96

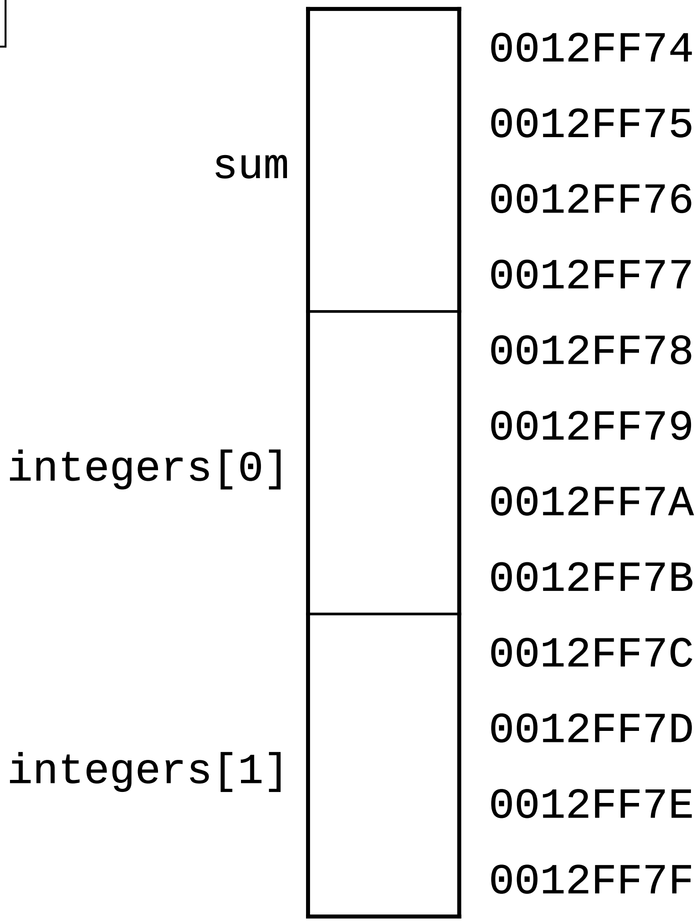
integer1 28

0012FF74
0012FF75
0012FF76
0012FF77
0012FF78
0012FF79
0012FF7A
0012FF7B
0012FF7C
0012FF7D
0012FF7E
0012FF7F

```
#include <iostream>
using namespace std;

int main()
{
   // array declarations
   int integers[ 2 ]; // an array
   int sum;

   cout << "Enter first integer: ";
   cin >> integers[ 0 ];

   cout << "Enter second integer: ";
   cin >> integers[ 1 ];

   sum = integers[ 0 ] + integers[ 1 ];

   cout << "Sum is " << sum << endl;
}
```

```
cout << "Enter first integer: ";
cin >> integers[ 0 ];
cout << "Enter second integer: ";
cin >> integers[ 1 ];
sum = integers[ 0 ] + integers[ 1 ];
cout << "Sum is " << sum << endl;
```

Memory



sum

integers[0]

integers[1]

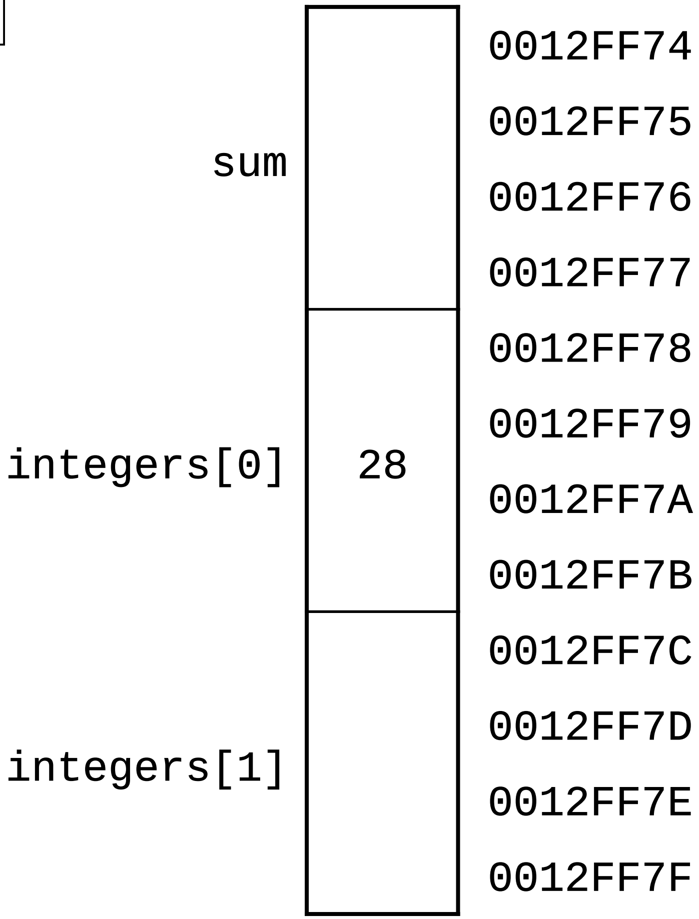
0012FF74
0012FF75
0012FF76
0012FF77
0012FF78
0012FF79
0012FF7A
0012FF7B
0012FF7C
0012FF7D
0012FF7E
0012FF7F

Output

```
Enter first integer: 28_
```

```
cout << "Enter first integer: ";
cin >> integers[ 0 ];
cout << "Enter second integer: ";
cin >> integers[ 1 ];
sum = integers[ 0 ] + integers[ 1 ];
cout << "Sum is " << sum << endl;
```

Memory



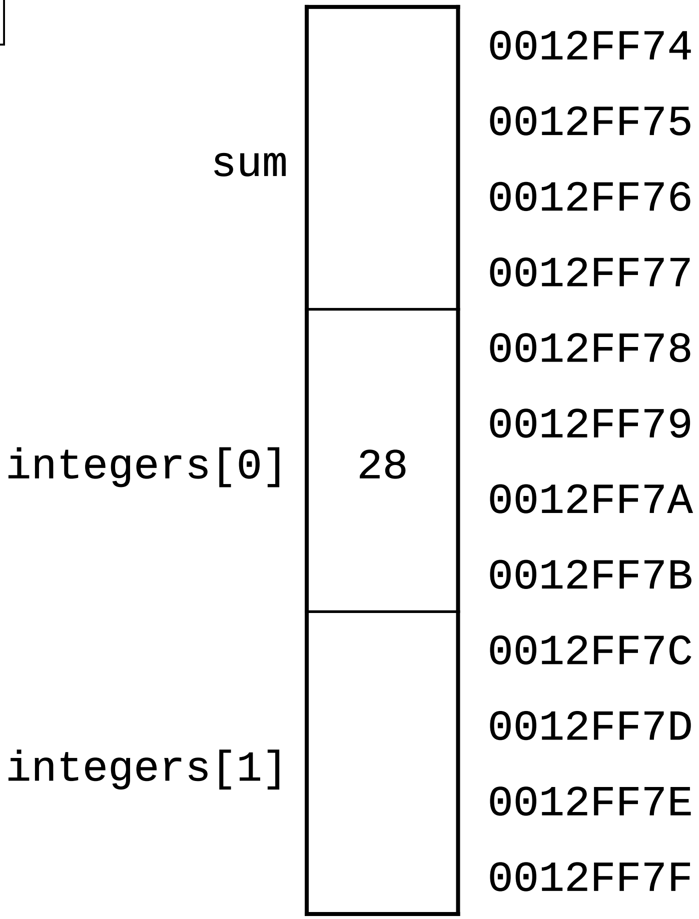
sum		0012FF74
		0012FF75
		0012FF76
		0012FF77
integers[0]	28	0012FF78
		0012FF79
		0012FF7A
		0012FF7B
integers[1]		0012FF7C
		0012FF7D
		0012FF7E
		0012FF7F

Output

```
Enter first integer: 28
_
```

```
cout << "Enter first integer: ";
cin >> integers[ 0 ];
cout << "Enter second integer: ";
cin >> integers[ 1 ];
sum = integers[ 0 ] + integers[ 1 ];
cout << "Sum is " << sum << endl;
```

Memory



sum

integers[0] 28

integers[1]

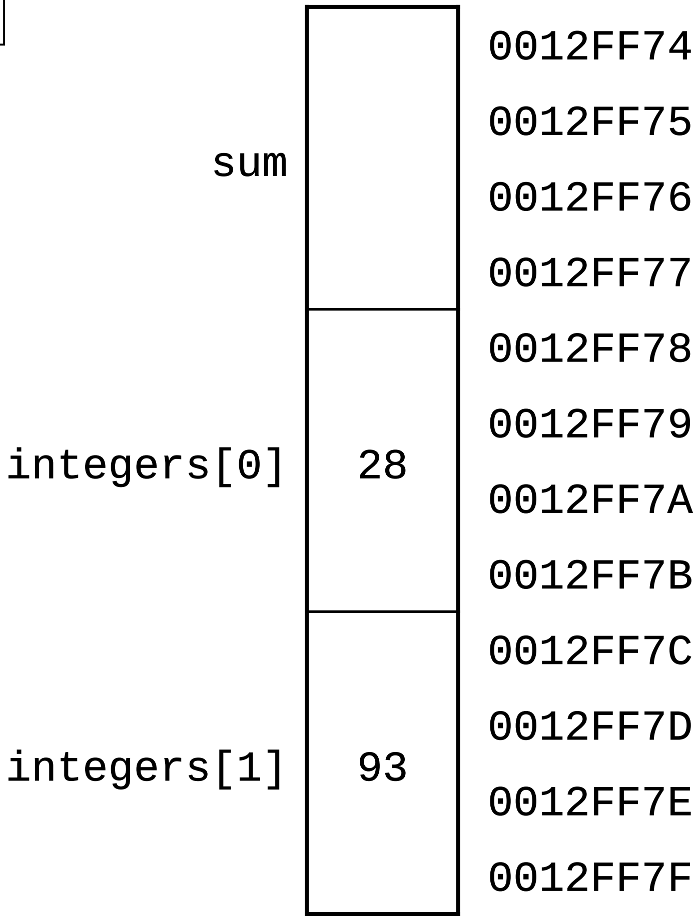
0012FF74
0012FF75
0012FF76
0012FF77
0012FF78
0012FF79
0012FF7A
0012FF7B
0012FF7C
0012FF7D
0012FF7E
0012FF7F

Output

```
Enter first integer: 28
Enter second integer: 93_
```

```
cout << "Enter first integer: ";
cin >> integers[ 0 ];
cout << "Enter second integer: ";
cin >> integers[ 1 ];
sum = integers[ 0 ] + integers[ 1 ];
cout << "Sum is " << sum << endl;
```

Memory

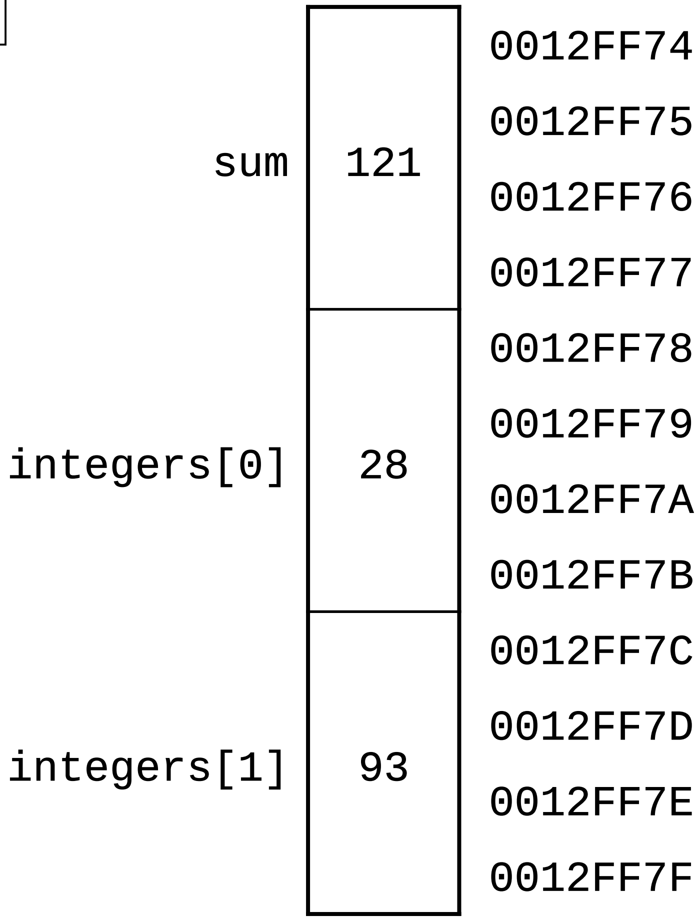


Output

```
Enter first integer: 28
Enter second integer: 93
_
```

```
cout << "Enter first integer: ";
cin >> integers[ 0 ];
cout << "Enter second integer: ";
cin >> integers[ 1 ];
sum = integers[ 0 ] + integers[ 1 ];
cout << "Sum is " << sum << endl;
```

Memory



sum 121

integers[0] 28

integers[1] 93

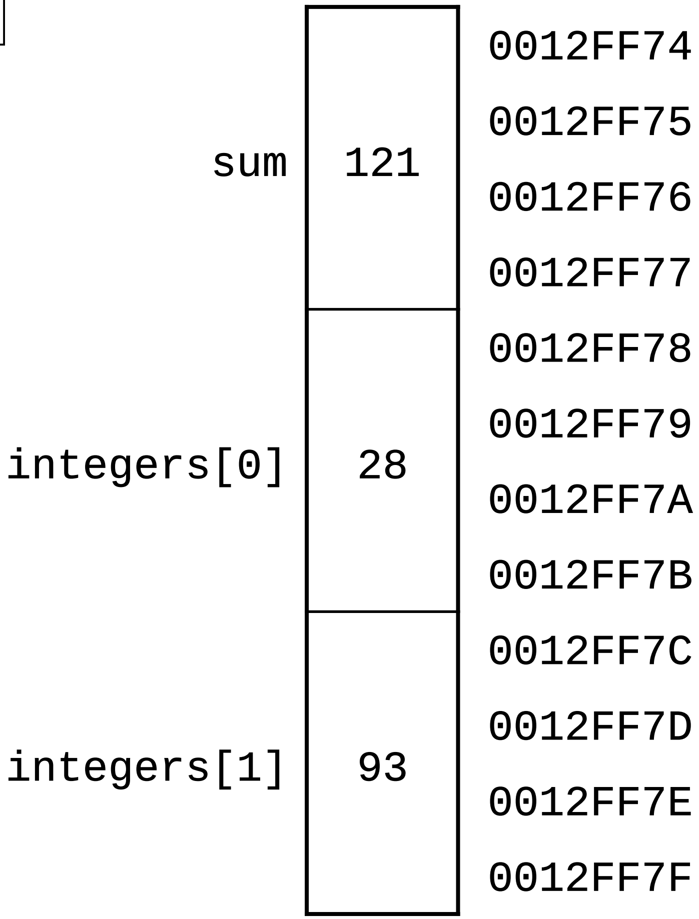
0012FF74
0012FF75
0012FF76
0012FF77
0012FF78
0012FF79
0012FF7A
0012FF7B
0012FF7C
0012FF7D
0012FF7E
0012FF7F

Output

```
Enter first integer: 28
Enter second integer: 93
_
```

```
cout << "Enter first integer: ";
cin >> integers[ 0 ];
cout << "Enter second integer: ";
cin >> integers[ 1 ];
sum = integers[ 0 ] + integers[ 1 ];
cout << "Sum is " << sum << endl;
```

Memory



sum 121

integers[0] 28

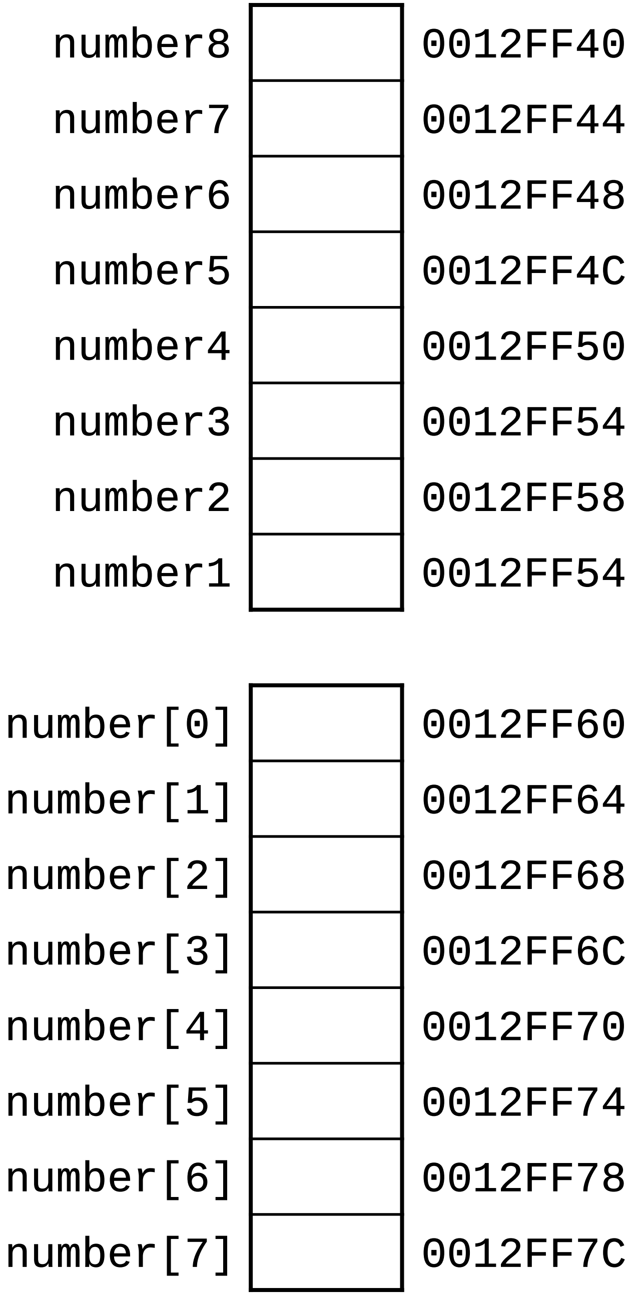
integers[1] 93

0012FF74
0012FF75
0012FF76
0012FF77
0012FF78
0012FF79
0012FF7A
0012FF7B
0012FF7C
0012FF7D
0012FF7E
0012FF7F

Output

```
Enter first integer: 28
Enter second integer: 93
Sum is 121

_
```

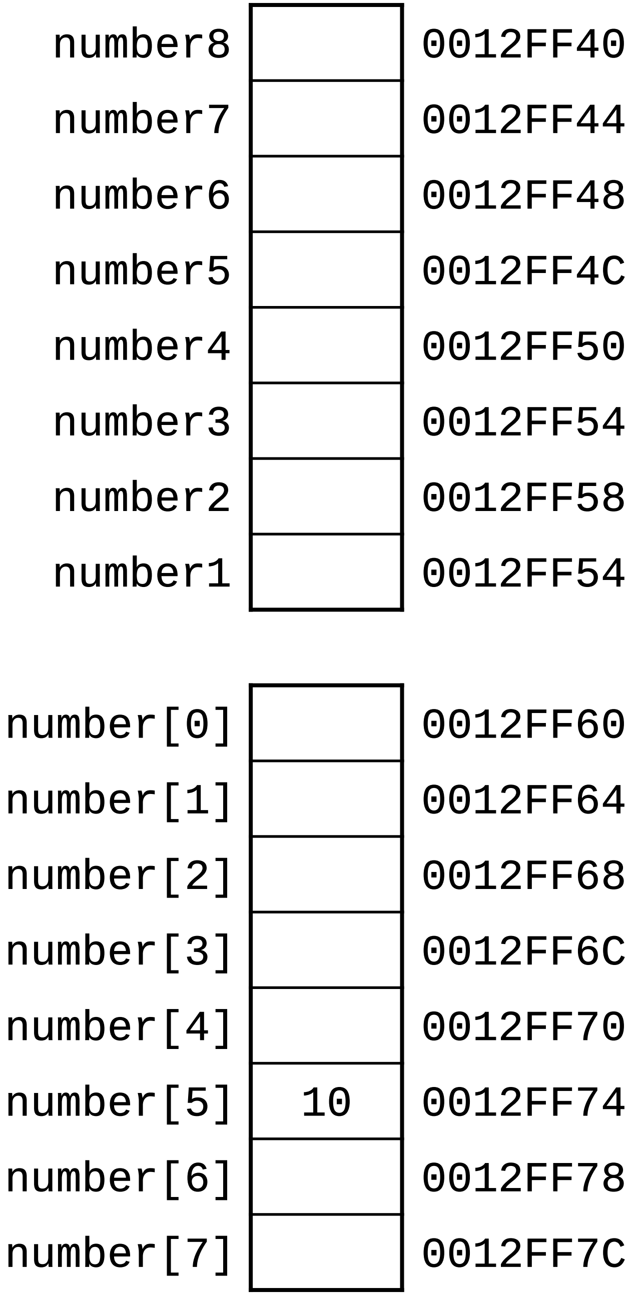
```
int number[ 8 ]; // eight integers

int number1; // first integer
int number2; // second integer
int number3; // third integer
int number4; // fourth integer
int number5; // fifth integer
int number6; // sixth integer
int number7; // seventh integer
int number8; // eighth integer

number[ 5 ] = 10;

number3 = 20;
```



number8 0012FF40
number7 0012FF44
number6 0012FF48
number5 0012FF4C
number4 0012FF50
number3 0012FF54
number2 0012FF58
number1 0012FF54

number[0] 0012FF60
number[1] 0012FF64
number[2] 0012FF68
number[3] 0012FF6C
number[4] 0012FF70
number[5] 0012FF74
number[6] 0012FF78
number[7] 0012FF7C

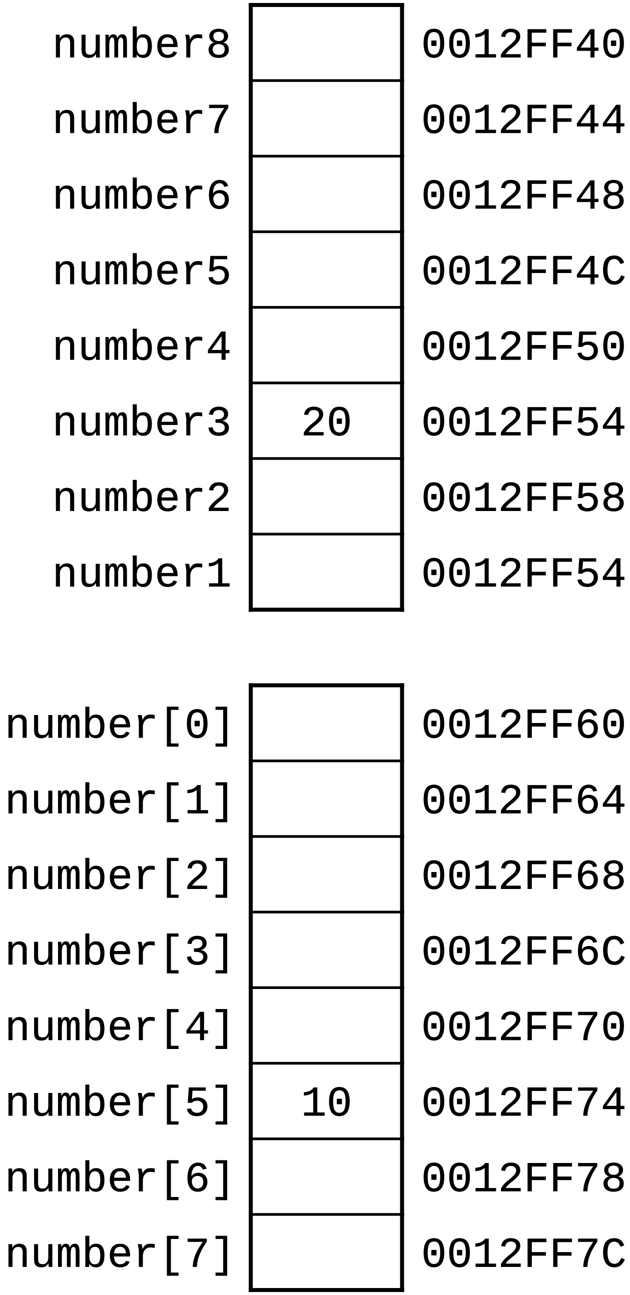
```
int number[ 8 ]; // eight integers

int number1; // first integer
int number2; // second integer
int number3; // third integer
int number4; // fourth integer
int number5; // fifth integer
int number6; // sixth integer
int number7; // seventh integer
int number8; // eighth integer

number[ 5 ] = 10;

number3 = 20;
```



number8		0012FF40
number7		0012FF44
number6		0012FF48
number5		0012FF4C
number4		0012FF50
number3		0012FF54
number2		0012FF58
number1		0012FF54

number[0]		0012FF60
number[1]		0012FF64
number[2]		0012FF68
number[3]		0012FF6C
number[4]		0012FF70
number[5]	10	0012FF74
number[6]		0012FF78
number[7]		0012FF7C

```
int number[ 8 ]; // eight integers

int number1; // first integer
int number2; // second integer
int number3; // third integer
int number4; // fourth integer
int number5; // fifth integer
int number6; // sixth integer
int number7; // seventh integer
int number8; // eighth integer

number[ 5 ] = 10;

number3 = 20;
```



number8		0012FF40
number7		0012FF44
number6		0012FF48
number5		0012FF4C
number4		0012FF50
number3	20	0012FF54
number2		0012FF58
number1		0012FF54

number[0]		0012FF60
number[1]		0012FF64
number[2]		0012FF68
number[3]		0012FF6C
number[4]		0012FF70
number[5]	10	0012FF74
number[6]		0012FF78
number[7]		0012FF7C

Arithmetic Operators

```
int main()
{
   int a, b;
   a = 7;
   b = 2;

   // printing the sum of a and b
   cout << "a + b = " << ( a + b ) << endl;

   // printing the difference of a and b
   cout << "a - b = " << ( a - b ) << endl;

   // printing the product of a and b
   cout << "a * b = " << ( a * b ) << endl;

   // printing the division of a by b
   cout << "a / b = " << ( a / b ) << endl;

   // printing the modulo of a by b
   cout << "a % b = " << ( a % b ) << endl;
}
```

```
a + b = 9
a - b = 5
a * b = 14
a / b = 3
a % b = 1
```

Initialization

```
int main()
{
   int a = 5; // initializer after equals sign
   int b( 6 ); // initializer in parenthesis
   int c{ 7 }; // initializer in braces
   int d = { 8 }; // initializer in braces
   int e{}; // initializer in braces

   cout << "a: " << a << endl;
   cout << "b: " << b << endl;
   cout << "c: " << c << endl;
   cout << "d: " << d << endl;
   cout << "e: " << e << endl;
   cout << "int(): " << int() << endl;
}
```

Output

```
a: 5
b: 6
c: 7
d: 8
e: 0
int(): 0
```

Initialization

```
int main()
{
   int a = int();
//   int b( int() ); // wrong
   int c{ int() };
   int d = { int() };
   int e;

   cout << "a: " << a << endl;
//   cout << "b: " << b << endl;
   cout << "c: " << c << endl;
   cout << "d: " << d << endl;
   e = int();
   cout << "e: " << e << endl;
   cout << "int(): " << int() << endl;
}
```

Output

```
a: 0
c: 0
d: 0
e: 0
int(): 0
```

if, if...else and Nested if...else

```
// Program to print positive number entered by the user
// If the user enters a negative number, it is skipped

#include <iostream>
using std::cout;
using std::cin;
using std::endl;

int main()
{
   int number;

   cout << "Enter an integer: ";
   cin >> number;

   // checks if the number is positive
   if( number > 0 )
      cout << "You entered a positive integer: " << number << endl;

   cout << "This statement is always executed.\n";
}
```

```
Enter an integer: 23
You entered a positive integer: 23
This statement is always executed.
```

```
Enter an integer: -7
This statement is always executed.
```

```cpp
// Program to check whether an integer is nonnegative or negative

#include <iostream>
using namespace std;

int main()
{
   int number;

   cout << "Enter an integer: ";
   cin >> number;

   if( number >= 0 )
      cout << "You entered a nonnegative integer: " << number << endl;
   else
      cout << "You entered a negative integer: " << number << endl;

   cout << "This line is always printed.\n";
}
```

```
Enter an integer: 23
You entered a positive integer: 23
This line is always printed.
```

```
Enter an integer: -7
You entered a negative integer: -7
This line is always printed.
```

```
// Program to check whether an integer is positive, negative or zero

#include <iostream>
using namespace std;

int main()
{
   int number;

   cout << "Enter an integer: ";
   cin >> number;

   if( number > 0 )
      cout << "You entered a positive integer: " << number << endl;
   else if( number < 0 )
      cout << "You entered a negative integer: " << number << endl;
   else
      cout << "You entered 0." << endl;

   cout << "This line is always printed.\n";
}
```

```
Enter an integer: 23
You entered a positive integer: 23
This line is always printed.
```

```
Enter an integer: -7
You entered a negative integer: -7
This line is always printed.
```

```
Enter an integer: 0
You entered 0.
This line is always printed.
```

```
// Program to check whether an integer is positive, negative or zero

#include <iostream>
using namespace std;
int main()
{
   int num;
   cout << "Enter an integer: ";
   cin >> num;
   if( num != 0 ) // outer if condition
      if( num > 0 ) // inner if condition
         cout << "The number is positive.\n";
      else // inner else condition
         cout << "The number is negative.\n";
   else // outer else condition
      cout << "The number is 0.\n";

   cout << "This line is always printed.\n";
}
```

Nested if . . . else Statements

```
if( a > 8 )
   if( b > 8 )
      cout << "a and b are > 8";
else
   cout << "a is <= 8";
```

The compiler interprets the above statement as

Nested if . . . else Statements

```
if( a > 8 )
   if( b > 8 )
      cout << "a and b are > 8";
else
   cout << "a is <= 8";
```

The compiler interprets the above statement as

```
if( a > 8 )
   if( b > 8 )
      cout << "a and b are > 8";
   else
      cout << "a is <= 8";
```

Nested if . . . else Statements

- To force the nested `if`…`else` statement to execute as intended, use:

```
if( a > 8 )
{
   if( b > 8 )
      cout << "a and b are > 8";
}
else
   cout << "a is <= 8";
```

- Braces (`{}`) indicate that the second `if` statement is in the body of the first `if` and that the `else` is associated with the first `if`.

Compound statements (Blocks)

- An example

```
if( grade >= 60 )
   cout << "Passed.\n";
else
{
   cout << "Failed.\n";
   cout << "You must take this course again.\n";
}
```

- Without the braces,

```
cout << "You must take this course again.\n";
```

  always executed.

Compound statements (Blocks)

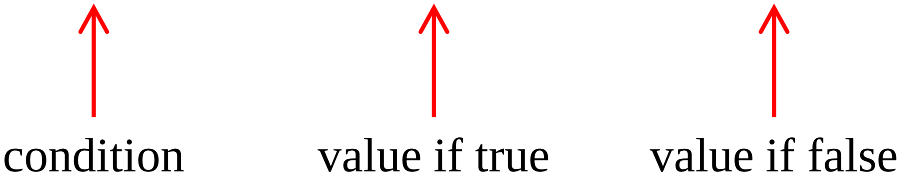
- The `if` selection statement expects only one statement in its body.
- Similarly, the `if` and `else` parts of an `if...else` statement each expect only one body statement.
- To include several statements in the body of an `if` or in either part of an `if...else`, enclose the statements in braces (`{` and `}`).
- A set of statements contained within a pair of braces is called a compound statement or a block.

C++ keywords

- alignas (C++11)
- alignof (C++11)
- and
- and_eq
- asm
- auto (1)
- bitand
- bitor
- bool
- break
- case
- catch
- char
- char8_t (C++20)
- char16_t (C++11)
- char32_t (C++11)
- class (1)
- compl
- concept (C++20)
- const
- consteval (C++20)
- constexpr (C++11)
- constinit (C++20)
- const_cast
- continue
- co_await (C++20)
- co_return (C++20)
- co_yield (C++20)
- decltype (C++11)
- default (1)
- delete (1)
- do
- double
- dynamic_cast
- else
- enum
- explicit
- export (1) (3)
- extern (1)
- false
- float
- for
- friend
- goto
- if
- inline (1)
- int
- long
- mutable (1)
- namespace
- new
- noexcept (C++11)
- not
- not_eq
- nullptr (C++11)
- operator
- or
- or_eq
- private
- protected
- public
- register (2)
- reinterpret_cast
- requires (C++20)
- return
- short
- signed
- sizeof (1)
- static
- static_assert(C++11)
- static_cast
- struct (1)
- switch
- template
- this (4)
- thread_local (C++11)
- throw
- true
- try
- typedef
- typeid
- typename
- union
- unsigned
- using (1)
- virtual
- void
- volatile
- wchar_t
- while
- xor
- xor_eq

Conditional operator (?:)

```
number >= 0 ? cout << "positive" : cout << "negative";

cout << ( number >= 0 ? "positive" : "negative" );
```



condition

value if true

value if false

Operators Precedence & Associativity Table

Operators Precedence & Associativity

Operator	Associativity
! ++ -- (post) ++ -- (pre) + - (unary)	Right to left
* / %	Left to right
+ - (binary)	Left to right
< <= > >=	Left to right
== !=	Left to right
&&	Left to right
||	Left to right
?:	Right to left
= *= /= %= += -=	Right to left

Assignment Operators

```
a = a + 7;   abbreviated to   a += 7;
b = b - 4;   abbreviated to   b -= 4;
c = c * 5;   abbreviated to   c *= 5;
d = d / 3;   abbreviated to   d /= 3;
e = e % 9;   abbreviated to   e %= 9;
```

Increment and Decrement Operators

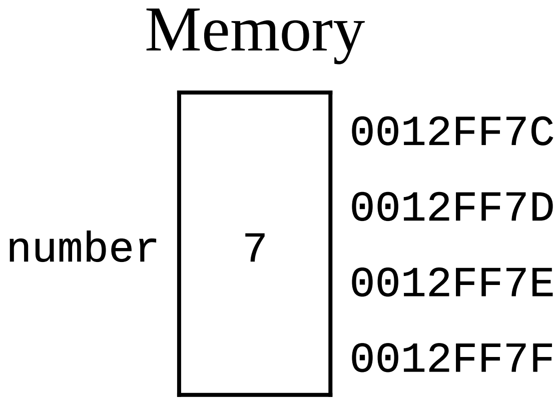
- Increment operator (`++`)
  - Increment a variable by one
  - `i++`
    - Same as `i += 1`
- Decrement operator (`--`)
  - Decrement a variable by one
  - `i--`
    - Same as `i -= 1`

Increment and Decrement Operators

- Preincrement
  - Variable changed before used in expression
    - Operator before variable (`++i` or `--i`)
- Postincrement
  - Incremented changed after expression
    - Operator after variable (`i++, i--`)

```cpp
int main()
{
   int number;

   number = 7;
   cout << number << endl;
   cout << number++ << endl;
   cout << number << endl;

   cout << endl;

   number = 7;
   cout << number << endl;
   cout << ++number << endl;
   cout << number << endl;
}
```

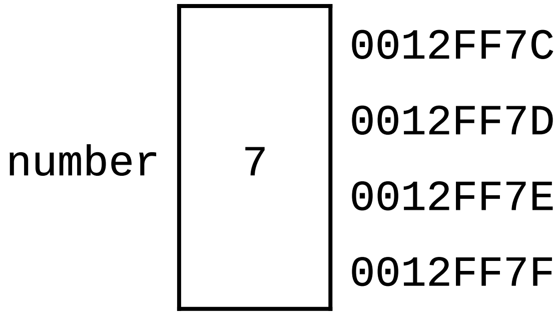
```
number = 7;
cout << number << endl;
cout << number++ << endl;
cout << number << endl;
```



Output

```
number = 7;
cout << number << endl;
cout << number++ << endl;
cout << number << endl;
```

Memory



number

7

0012FF7C

0012FF7D

0012FF7E

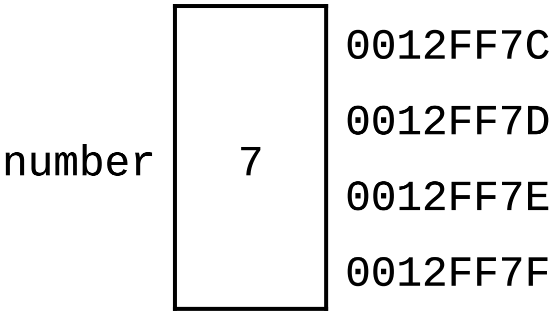
0012FF7F

Output

```
7
```

```
number = 7;
cout << number << endl;
cout << number++ << endl;
cout << number << endl;
```

Memory

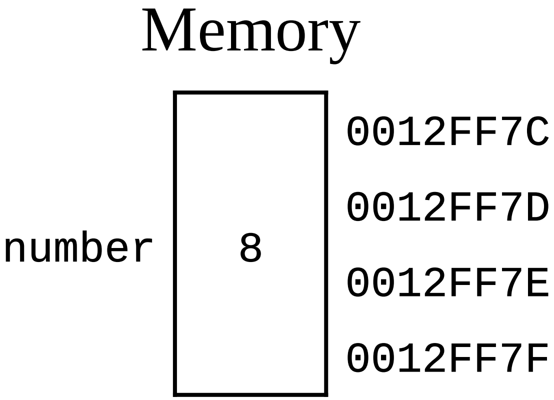


Output

```
7
7
```

```
number = 7;
cout << number << endl;
cout << number++ << endl;
cout << number << endl;
```

Memory



number 8

0012FF7C

0012FF7D

0012FF7E

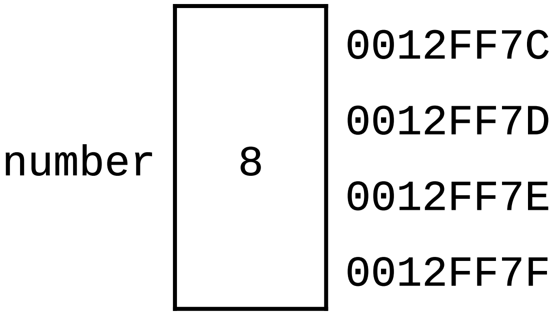
0012FF7F

Output

```
7
7
```

```
number = 7;
cout << number << endl;
cout << number++ << endl;
cout << number << endl;
```

Memory



number

8

0012FF7C

0012FF7D

0012FF7E

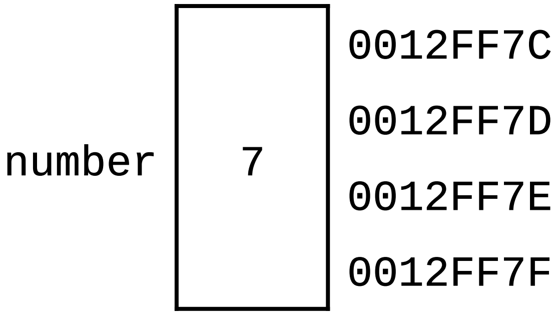
0012FF7F

Output

```
7
7
8
```

```
number = 7;
cout << number << endl;
cout << ++number << endl;
cout << number << endl;
```

Memory

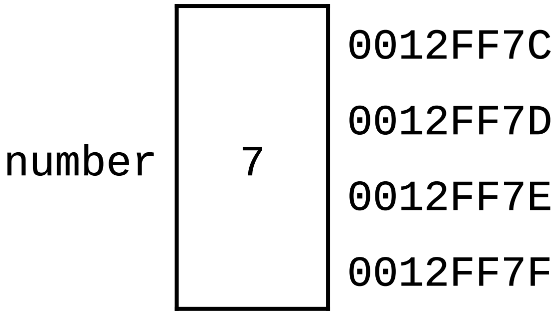


number
7
0012FF7C
0012FF7D
0012FF7E
0012FF7F

Output

```
number = 7;
cout << number << endl;
cout << ++number << endl;
cout << number << endl;
```

Memory



number 7

0012FF7C

0012FF7D

0012FF7E

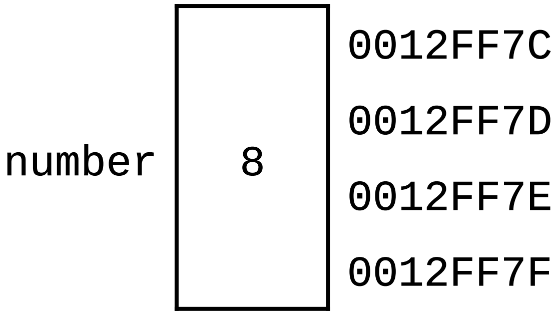
0012FF7F

Output

```
7
```

```
number = 7;
cout << number << endl;
cout << ++number << endl;
cout << number << endl;
```

Memory



number

8

0012FF7C

0012FF7D

0012FF7E

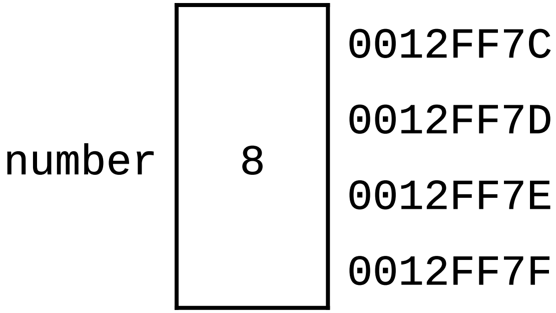
0012FF7F

Output

```
7
```

```
number = 7;
cout << number << endl;
cout << ++number << endl;
cout << number << endl;
```

Memory



number 8

0012FF7C
0012FF7D
0012FF7E
0012FF7F

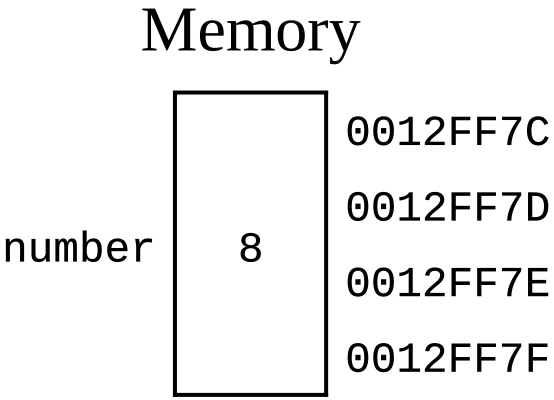
Output

```
7
8
```

```
number = 7;
cout << number << endl;
cout << ++number << endl;
cout << number << endl;
```

Output

```
7
8
8
```



Memory

Increment and Decrement Operators

- When you increment (`++`) or decrement (`--`) a variable in a statement by itself, the preincrement and postincrement forms have the same effect

```
++number;
cout << number;
```

and

```
number++;
cout << number;
```

print the same value.

Operators Precedence & Associativity

Operator	Associativity
! ++ -- (post) ++ -- (pre) + - (unary)	Right to left
* / %	Left to right
+ - (binary)	Left to right
<< >>	Left to right
< <= > >=	Left to right
== !=	Left to right
&&	Left to right
||	Left to right
?:	Right to left
= *= /= %= += -=	Right to left

Data types
long double (8 bytes) 絕對值範圍大約是 $2.2 \cdot 10^{-308}$ ~ $1.8 \cdot 10^{308}$
double (8 bytes) 絕對值範圍大約是 $2.2 \cdot 10^{-308}$ ~ $1.8 \cdot 10^{308}$
float (4 bytes) 絕對值範圍大約是 $1.2 \cdot 10^{-38}$ ~ $3.4 \cdot 10^{38}$
unsigned long long int (unsigned long long) (8 bytes) 0 ~ 2^{64}-1
long long int (long long) (8 bytes) -2^{63} ~ 2^{63}-1
unsigned long int (unsigned long) (4 bytes) 0 ~ 2^{32}-1
long int (long) (4 bytes) -2^{31} ~ 2^{31}-1 (2147483647)
unsigned int (unsigned) (4 bytes) 0 ~ 2^{32}-1 (4294967295)
int (4 bytes) -2^{31} ~ 2^{31}-1 (2147483647)
unsigned short int (unsigned short) (2) 0 ~ 2^{16}-1 (65535)
short int (short) (2 bytes) -2^{15} ~ 2^{15}-1 (32767)
unsigned char (1 byte) 0 ~ 2^{8}-1 (0 ~ 255)
char (1 byte) -2^{7} ~ 2^{7}-1 (-128 ~ 127)
bool (1 byte) (false becomes 0, true becomes 1)

binary representation	int	
01111111 11111111 11111111 11111111	2147483647	$2^{31} - 1$
01111111 11111111 11111111 11111110	2147483646	$2^{31} - 2$
01111111 11111111 11111111 11111100	2147483645	$2^{31} - 3$
....	....	
00000000 00000000 00000000 00000011	3	
00000000 00000000 00000000 00000010	2	
00000000 00000000 00000000 00000001	1	
00000000 00000000 00000000 00000000	0	
11111111 11111111 11111111 11111111	-1	
11111111 11111111 11111111 11111110	-2	
11111111 11111111 11111111 11111101	-3	
....	....	
10000000 00000000 00000000 00000010	-2147483646	$-2^{31} + 2$
10000000 00000000 00000000 00000001	-2147483647	$-2^{31} + 1$
10000000 00000000 00000000 00000000	-2147483648	-2^{31}

binary representation	unsigned int	
11111111 11111111 11111111 11111111	4294967295	$2^{32} - 1$
11111111 11111111 11111111 11111110	4294967294	$2^{32} - 2$
11111111 11111111 11111111 11111101	4294967293	$2^{32} - 3$
....	....	
10000000 00000000 00000000 00000010	2147483650	$2^{31} + 2$
10000000 00000000 00000000 00000001	2147483649	$2^{31} + 1$
10000000 00000000 00000000 00000000	2147483648	2^{31}
01111111 11111111 11111111 11111111	2147483647	$2^{31} - 1$
01111111 11111111 11111111 11111110	2147483646	$2^{31} - 2$
01111111 11111111 11111111 11111100	2147483645	$2^{31} - 3$
....	....	
00000000 00000000 00000000 00000011	3	
00000000 00000000 00000000 00000010	2	
00000000 00000000 00000000 00000001	1	
00000000 00000000 00000000 00000000	0	

binary representation	int	unsigned int
01111111 11111111 11111111 11111111	2147483647	2147483647
01111111 11111111 11111111 11111110	2147483646	2147483646
01111111 11111111 11111111 11111100	2147483645	2147483645
....	....	....
00000000 00000000 00000000 00000011	3	3
00000000 00000000 00000000 00000010	2	2
00000000 00000000 00000000 00000001	1	1
00000000 00000000 00000000 00000000	0	0
11111111 11111111 11111111 11111111	-1	4294967295
11111111 11111111 11111111 11111110	-2	4294967294
11111111 11111111 11111111 11111101	-3	4294967293
....	....	....
10000000 00000000 00000000 00000010	-2147483646	2147483650
10000000 00000000 00000000 00000001	-2147483647	2147483649
10000000 00000000 00000000 00000000	-2147483648	2147483648

hexadecimal	long long int	unsigned long long
7FFFFFFFFFFFFFFF	9223372036854775807	9223372036854775807
7FFFFFFFFFFFFFFE	9223372036854775806	9223372036854775806
7FFFFFFFFFFFFFFD	9223372036854775805	9223372036854775805
....	....	....
0000000000000003	3	3
0000000000000002	2	2
0000000000000001	1	1
0000000000000000	0	0
FFFFFFFFFFFFFFFF	-1	18446744073709551615
FFFFFFFFFFFFFFFE	-2	18446744073709551614
FFFFFFFFFFFFFFFD	-3	18446744073709551613
....	....	....
8000000000000002	-9223372036854775806	9223372036854775810
8000000000000001	-9223372036854775807	9223372036854775809
8000000000000000	-9223372036854775808	9223372036854775808

Maximum Finding

```
int main()
{
   int number1;
   int number2;

   cin >> number1;
   cin >> number2;

   if(                    )
                     ;
   else
                     ;
}
```

Maximum Finding

```
int main()
{
   int number1;
   int number2;

   cin >> number1;
   cin >> number2;

   if( number1 >= number2 )
                     ;
   else
                     ;
}
```

Maximum Finding

```
int main()
{
   int number1;
   int number2;

   cin >> number1;
   cin >> number2;

   if( number1 >= number2 )
      cout << number1;
   else
                     ;
}
```

Maximum Finding

```
int main()
{
   int number1;
   int number2;

   cin >> number1;
   cin >> number2;

   if( number1 >= number2 )
      cout << number1;
   else
      cout << number2;
}
```

Maximum Finding

```
int main()
{
   int number1;
   int number2;
   int number3;
   cin >> number1;
   cin >> number2;
   cin >> number3;
   if(                     )
      if(                     )
                         ;
      else
                         ;
   else




}
```

Maximum Finding

```
int main()
{
   int number1;
   int number2;
   int number3;
   cin >> number1;
   cin >> number2;
   cin >> number3;
   if( number1 >= number2 )
      if(                    )
                        ;
      else
                        ;
   else




}
```

Maximum Finding

```
int main()
{
   int number1;
   int number2;
   int number3;
   cin >> number1;
   cin >> number2;
   cin >> number3;
   if( number1 >= number2 )
      if( number1 >= number3 )
                        ;
      else
                        ;
   else





}
```

Maximum Finding

```
int main()
{
   int number1;
   int number2;
   int number3;
   cin >> number1;
   cin >> number2;
   cin >> number3;
   if( number1 >= number2 )
      if( number1 >= number3 )
         cout << number1;
      else
                        ;
   else





}
```

Maximum Finding

```
int main()
{
   int number1;
   int number2;
   int number3;
   cin >> number1;
   cin >> number2;
   cin >> number3;
   if( number1 >= number2 )
      if( number1 >= number3 )
         cout << number1;
      else
         cout << number3;
   else





}
```

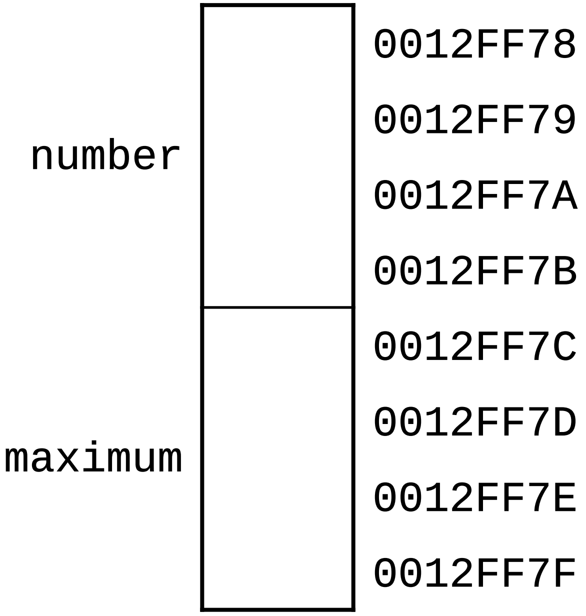
Maximum Finding

```
int main()
{
   int number1;
   int number2;
   int number3;
   cin >> number1;
   cin >> number2;
   cin >> number3;
   if( number1 >= number2 )
      if( number1 >= number3 )
         cout << number1;
      else
         cout << number3;
   else
      if( number2 >= number3 )
         cout << number2;
      else
         cout << number3;
}
```

```
int main()
{
  int number1;
  int number2;
  int number3;
  int number4;
  cin >> number1;
  cin >> number2;
  cin >> number3;
  cin >> number4;
  if( number1 >= number2 )
     if( number1 >= number3 )
        if( number1 >= number4 )
           cout << number1;
        else
           cout << number4;
     else
        if( number3 >= number4 )
           cout << number3;
        else
           cout << number4;
  else
     if( number2 >= number3 )
        if( number2 >= number4 )
           cout << number2;
        else
           cout << number4;
     else
        if( number3 >= number4 )
           cout << number3;
        else
           cout << number4;
}
```

Maximum Finding

```
cin >> maximum;
cin >> number;
```



number

maximum

0012FF78

0012FF79

0012FF7A

0012FF7B

0012FF7C

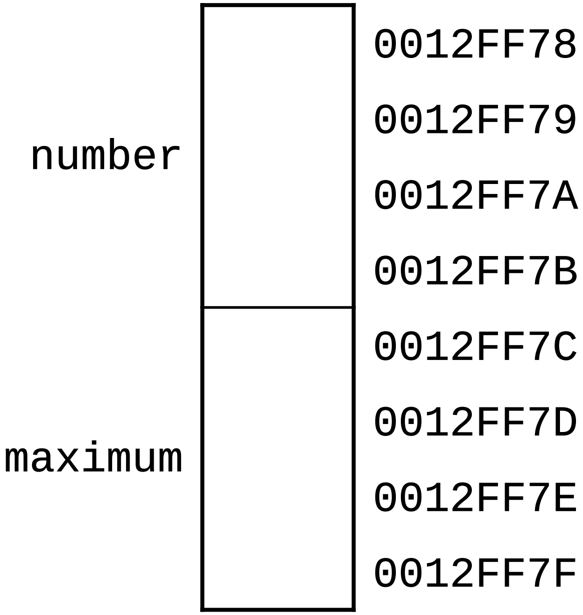
0012FF7D

0012FF7E

0012FF7F

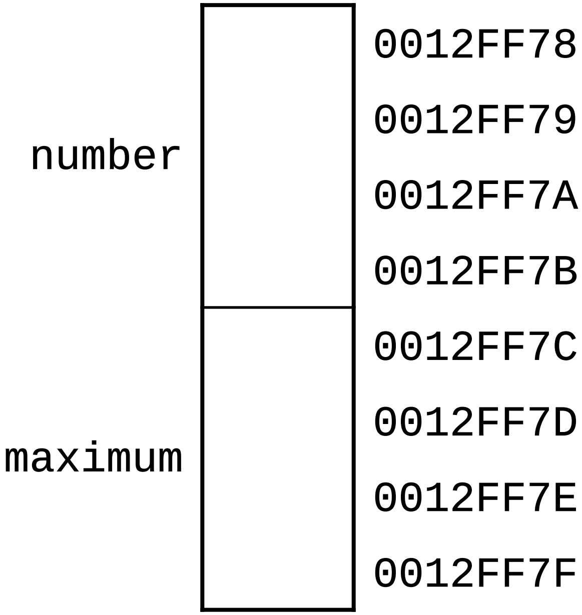
Maximum Finding

```
cin >> maximum;
cin >> number;
if( number > maximum )
```



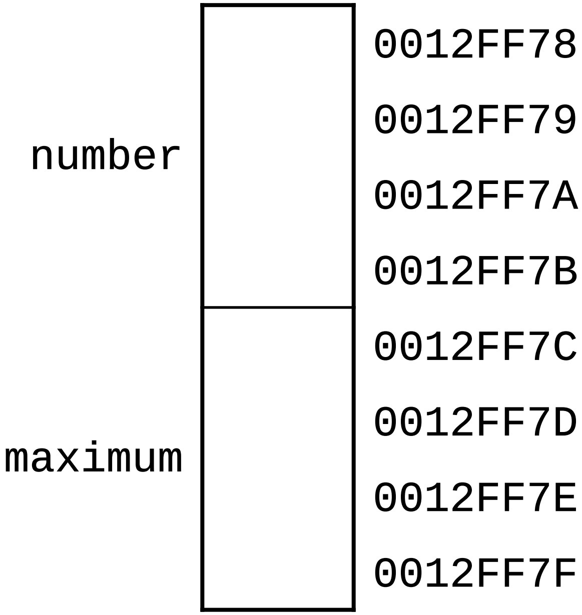
Maximum Finding

```
cin >> maximum;
cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
```



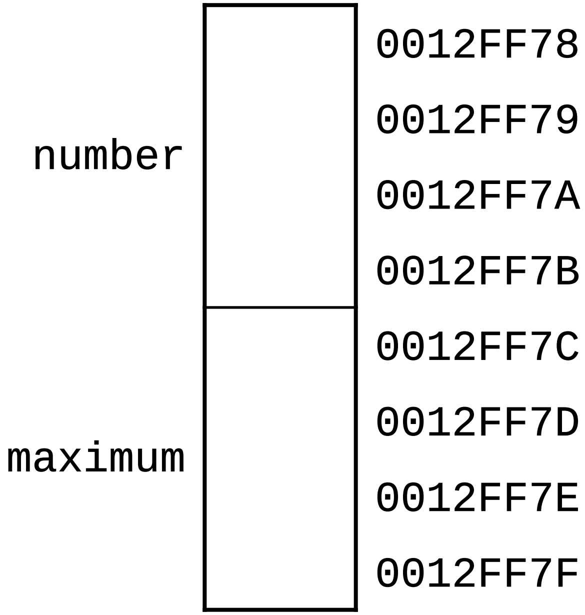
Maximum Finding

```
cin >> maximum;
cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
```



Maximum Finding

```
cin >> maximum;
cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
```



number

maximum

0012FF78

0012FF79

0012FF7A

0012FF7B

0012FF7C

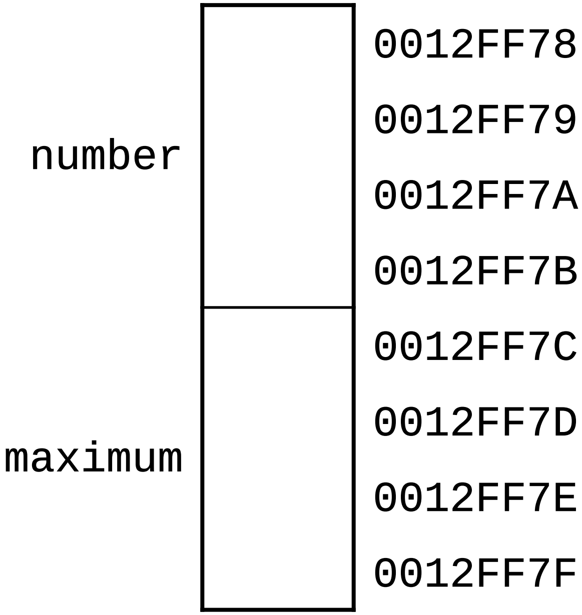
0012FF7D

0012FF7E

0012FF7F

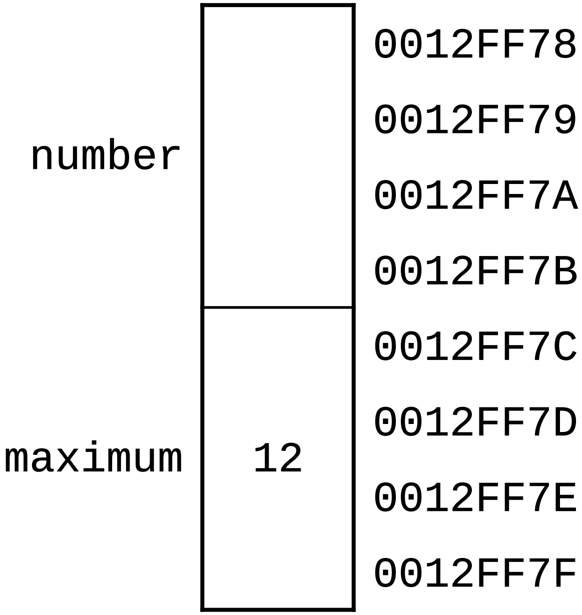
Maximum Finding

```
cin >> maximum;
cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;
```



Maximum Finding

```
cin >> maximum;
cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;
```



number

maximum

12

0012FF78

0012FF79

0012FF7A

0012FF7B

0012FF7C

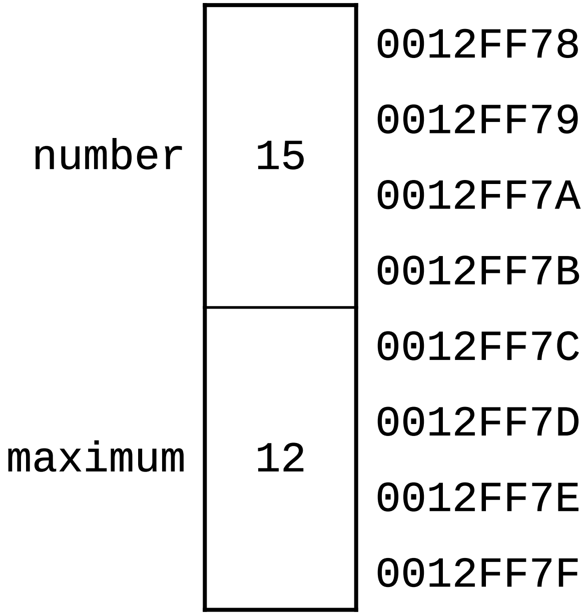
0012FF7D

0012FF7E

0012FF7F

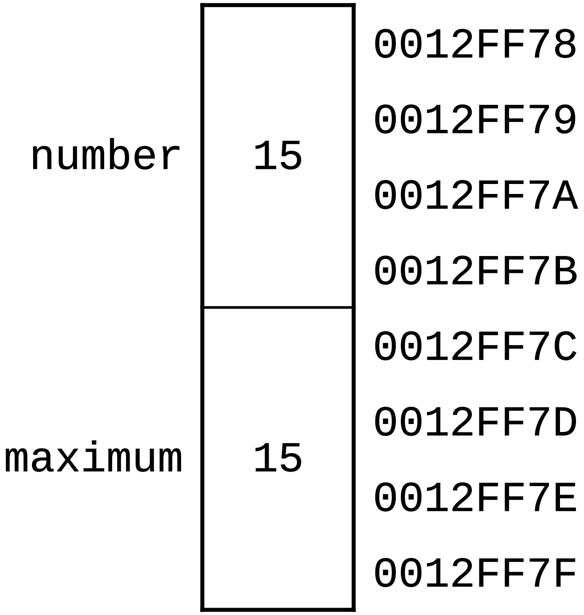
Maximum Finding

```
cin >> maximum;
cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;
```



Maximum Finding

```
cin >> maximum;
cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;
```



number 15

maximum 15

0012FF78

0012FF79

0012FF7A

0012FF7B

0012FF7C

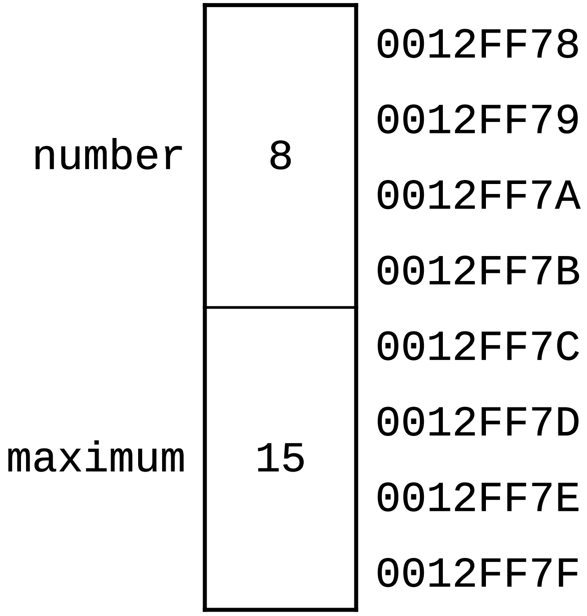
0012FF7D

0012FF7E

0012FF7F

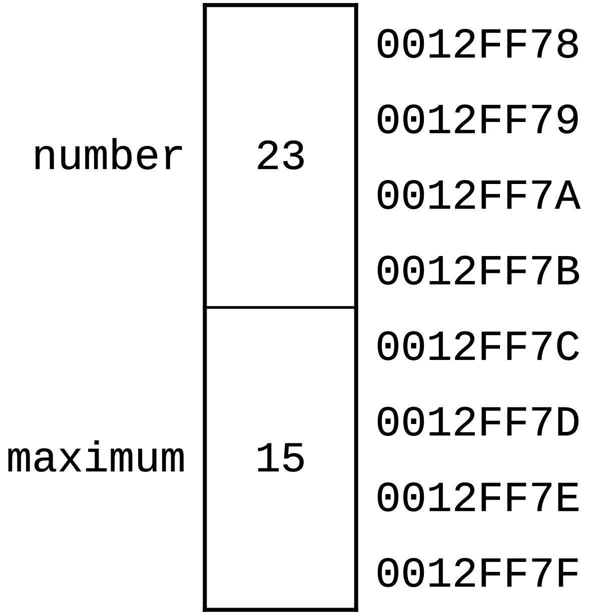
Maximum Finding

```
cin >> maximum;
cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;
```



Maximum Finding

```
cin >> maximum;
cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;
```



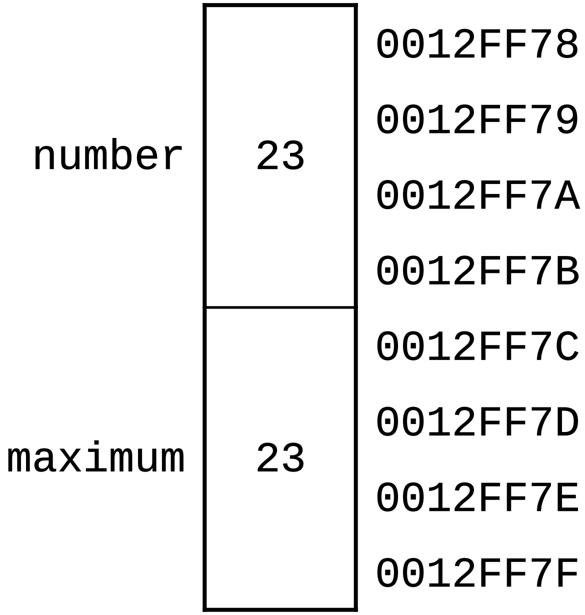
number 23

maximum 15

0012FF78
0012FF79
0012FF7A
0012FF7B
0012FF7C
0012FF7D
0012FF7E
0012FF7F

Maximum Finding

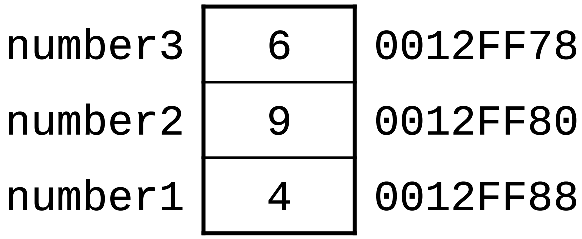
```
cin >> maximum;
cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;

cin >> number;
if( number > maximum )
    maximum = number;
```



Introduction to Functions

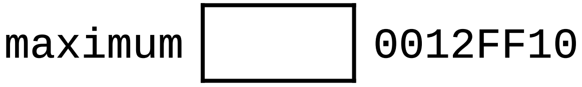
```
int main()
{
   int number1;
   int number2;
   int number3;

   cout << "Enter three numbers: ";
   cin >> number1 >> number2 >> number3;

   int maximum = number1; // assume number1 is largest

   if( number2 > maximum ) // if number2 is larger,
      maximum = number2; // assign number2 to maximum

   if( number3 > maximum ) // if number3 is larger,
      maximum = number3; // assign number3 to maximum

   cout << "Maximum is: " << maximum << endl;
}
```

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   max();
}
```



number3	6	0012FF78
number2	9	0012FF80
number1	4	0012FF88

```
int max()
{
   int maximum = number1;
   if( number2 > maximum )
      maximum = number2;
   if( number3 > maximum )
      maximum = number3;
   cout << maximum << endl;
}
```



maximum		0012FF10

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int maximum = number1;
   if( number2 > maximum )
      maximum = number2;
   if( number3 > maximum )
      maximum = number3;
   cout << maximum << endl;
}
```

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max << endl;
}

int maximum( int number1, int number2, int number3 )
{
   int max = number1;
   if( number2 > max )
      max = number2;
   if( number3 > max )
      max = number3;
   return max;
}
```

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int maximum = number1;
   if( number2 > maximum )
      maximum = number2;
   if( number3 > maximum )
      maximum = number3;
   cout << maximum << endl;
}
```

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max << endl;
}

int maximum( int a, int b, int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

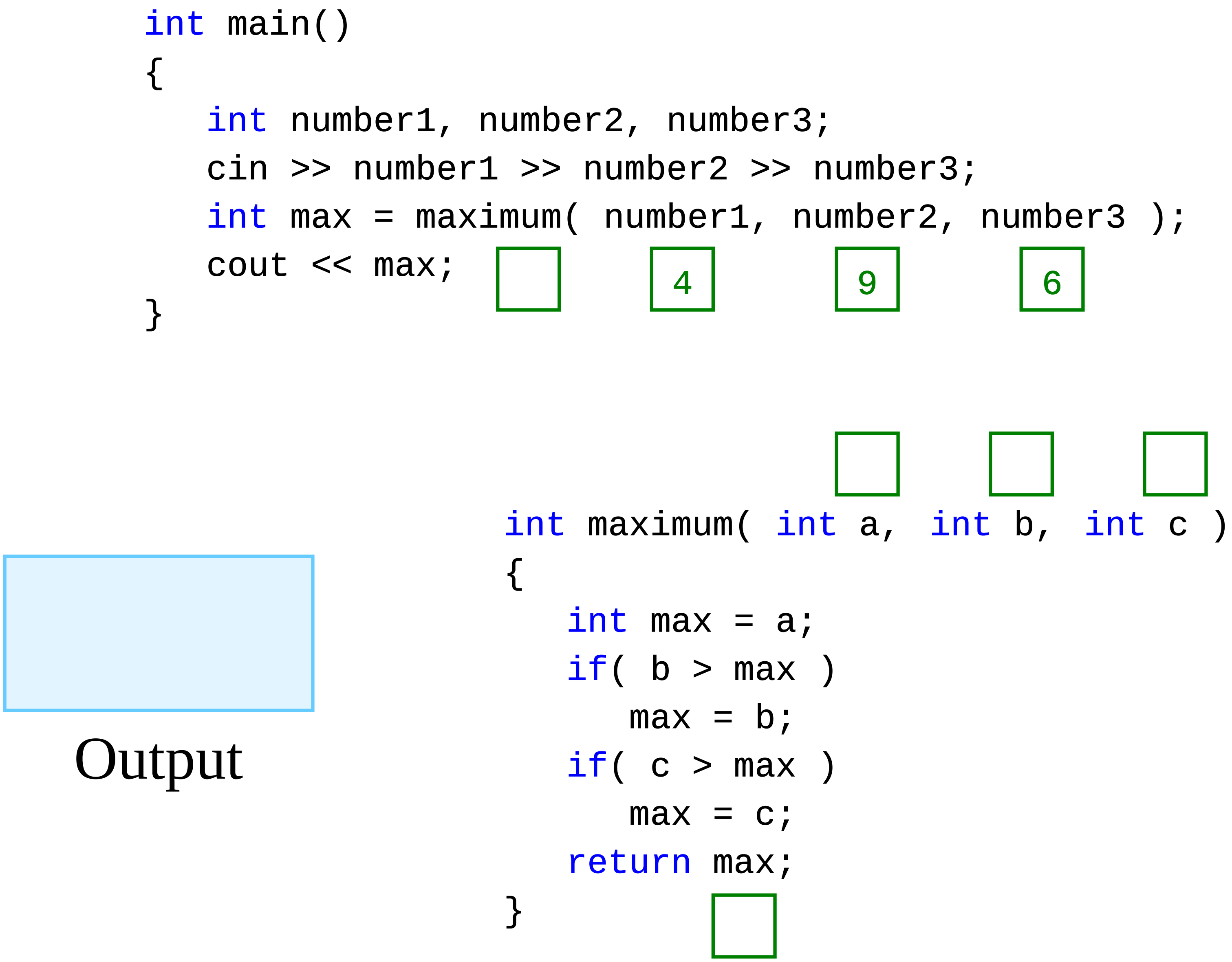
```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int maximum = number1;
   if( number2 > maximum )
      maximum = number2;
   if( number3 > maximum )
      maximum = number3;
   cout << maximum << endl;
}
```

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max;
}
```

4 9 6

```
int maximum( int a, int b, int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

Output



```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max;
}
```

```
int maximum( int a, int b, int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

Output

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max;
}
```

□ 4 9 6

4 9 6

```
int maximum( int a, int b, int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

9

Output

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max;
}
```

9 4 9 6

Output

```
int maximum( int a, int b, int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max;   9      4       9       6
}
```

```
9
```

Output

```
int maximum( int a, int b, int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

Output

		0012FF6C
max		0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max << endl;
}

int maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

Output

		0012FF6C
max		0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max << endl;
}

int maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

max		0012FF10
a		0012FF14
b		0012FF18
c		0012FF1C

Output

		0012FF6C
max		0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max << endl;
}

int maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

max	9	0012FF10
a	4	0012FF14
b	9	0012FF18
c	6	0012FF1C

Output

	9	0012FF6C
max		0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max << endl;
}

int maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

max	9	0012FF10
a	4	0012FF14
b	9	0012FF18
c	6	0012FF1C

Output

```
9
```

	9	0012FF6C
max	9	0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max << endl;
}

int maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

max	9	0012FF10
a	4	0012FF14
b	9	0012FF18
c	6	0012FF1C

Output

		0012FF6C
max		0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max << endl;
}

int maximum( int number1, int number2, int number3 )
{
   int max = number1;
   if( number2 > max )
      max = number2;
   if( number3 > max )
      max = number3;
   return max;
}
```

Output

		0012FF6C
max		0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max << endl;
}

int maximum( int number1, int number2, int number3 )
{
   int max = number1;
   if( number2 > max )
      max = number2;
   if( number3 > max )
      max = number3;
   return max;
}
```

max		0012FF10
number1		0012FF14
number2		0012FF18
number3		0012FF1C

Output

		0012FF6C
max		0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max << endl;
}

int maximum( int number1, int number2, int number3 )
{
   int max = number1;
   if( number2 > max )
      max = number2;
   if( number3 > max )
      max = number3;
   return max;
}
```

max	9	0012FF10
number1	4	0012FF14
number2	9	0012FF18
number3	6	0012FF1C

Output

	9	0012FF6C
max		0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max << endl;
}

int maximum( int number1, int number2, int number3 )
{
   int max = number1;
   if( number2 > max )
      max = number2;
   if( number3 > max )
      max = number3;
   return max;
}
```

max	9	0012FF10
number1	4	0012FF14
number2	9	0012FF18
number3	6	0012FF1C

Output

```
9
```

	9	0012FF6C
max	9	0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   int max = maximum( number1, number2, number3 );
   cout << max << endl;
}

int maximum( int number1, int number2, int number3 )
{
   int max = number1;
   if( number2 > max )
      max = number2;
   if( number3 > max )
      max = number3;
   return max;
}
```

max	9	0012FF10
number1	4	0012FF14
number2	9	0012FF18
number3	6	0012FF1C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   cout << maximum( number1, number2, number3 ) << endl;
}

int maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

Output

		0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   cout << maximum( number1, number2, number3 ) << endl;
}

int maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

Output

		0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   cout << maximum( number1, number2, number3 ) << endl;
}

int maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

max		0012FF10
a		0012FF14
b		0012FF18
c		0012FF1C

Output

		0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   cout << maximum( number1, number2, number3 ) << endl;
}

int maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

max	9	0012FF10
a	4	0012FF14
b	9	0012FF18
c	6	0012FF1C

Output

```
9
```

	9	0012FF70
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   cout << maximum( number1, number2, number3 ) << endl;
}

int maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   return max;
}
```

max	9	0012FF10
a	4	0012FF14
b	9	0012FF18
c	6	0012FF1C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   maximum( number1, number2, number3 );
}

void maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   cout << max << endl;
}
```

Output

number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   maximum( number1, number2, number3 );
}

void maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   cout << max << endl;
}
```

Output

number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   maximum( number1, number2, number3 );
}

void maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   cout << max << endl;
}
```

max		0012FF10
a		0012FF14
b		0012FF18
c		0012FF1C

Output

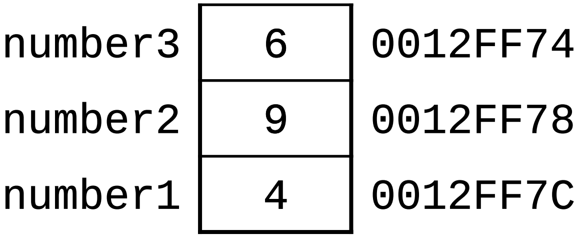
number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   maximum( number1, number2, number3 );
}

void maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   cout << max << endl;
}
```

max	9	0012FF10
a	4	0012FF14
b	9	0012FF18
c	6	0012FF1C

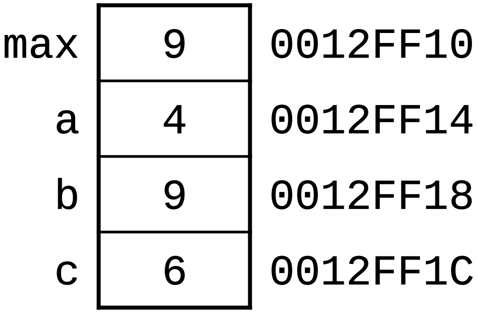
Output

```
9
```



number3	6	0012FF74
number2	9	0012FF78
number1	4	0012FF7C

```
int main()
{
   int number1, number2, number3;
   cin >> number1 >> number2 >> number3;
   maximum( number1, number2, number3 );
}

void maximum( int a,  int b,  int c )
{
   int max = a;
   if( b > max )
      max = b;
   if( c > max )
      max = c;
   cout << max << endl;
}
```



max	9	0012FF10
a	4	0012FF14
b	9	0012FF18
c	6	0012FF1C